ANO II · NÚMERO 16 – Cr$ 120,00

SPRITES E ROTINAS DE VÍDEO

POR DENTRO DA INTERFACE DE DRIVE

ANÁLISES: LADY 80 E ASTEX

COBRA'S ARK

GUN FRIGHT

Via Aéreo · MANAUS e BOA VISTA · Cr$ 1

ÁGUIA INFORMÁTICA LTDA.
AV. N. S. DE COPACABANA 605/804
COPACABANA
22040 – RIO DE JANEIRO – RJ
TELEFONE: PABX (021) 235-3541
TELEX: 21.21717 KPUR BR

DIRETOR RESPONSÁVEL
GONÇALO R. F. MURTEIRA

DIRETOR COMERCIAL
JOSÉ IDEMAR A. NASCIMENTO

ADMINISTRAÇÃO
MARCIA HAUCH DE CASTRO

PUBLICIDADE
JOSÉ NILTON BARROS DA SILVA

ASSINATURAS
MONICA VICENTE

JORNALISTA RESPONSÁVEL
DOLAR TANUS
REGISTRO 430-RS

COLABORADORES
PAULO MARQUES FIGUEIRA
SERGIO GUY PINHEIRO ELIAS
PAULO ROBERTO PINHEIRO ELIAS
BRUNO MARRUT
JULIO VELLOSO
SERGIO DURIC CALHEIROS
GUILHERME A. L. DA SILVA
ANDRÉ L. A. SANTOS

REVISÃO DE TEXTO
LAURA MARIA PINTO CERSOSIMO

CAPA
MILTON MEIER JR.

ARTE FINAL
WELLINGTON SILVARES

FOTOLITOS
PROJETA STUDIO GRÁFICO

COMPOSIÇÃO
STUDIO MORAES

IMPRESSÃO
EDITORA LORO

DISTRIBUIÇÃO
FERNANDO CHINAGLIA
DISTRIBUIDORA S.A.

EDITORIAL

A revista CPU passou por uma série de mudanças, tanto na capa como na disposição dos textos nas páginas interiores. As mudanças foram realizadas com o objetivo de atender uma série de sugestões que nos foram enviadas.

Para o próximo número, mais uma série de modificações serão introduzidas. O espaço destinado às colaborações enviadas pelos leitores, que muitas vezes são umas poucas linhas mas que em muito poderão ajudar outros usuários de MSX, será aumentado, sem prejudicar o espaço já destinado aos artigos técnicos, análise de software e hardware, programas, mapas de jogos, etc. Outras seções, solicitadas pelos leitores, também serão iniciadas.

Neste número de CPU analisamos a impressora Lady-80, da Elgin, que oferece um leque bem variado de opções para o usuário, além de uma qualidade de impressão excelente. Se você está pensando em adquirir este periférico, vale a pena ler esta análise.

O mercado de software para MSX tem andado bastante agitado e alguns lançamentos de programas nacionais já estão previstos para breve. Um deles, desenvolvido por um dos colaboradores da revista e que será lançado por uma softhouse do Rio, fará muitos pensarem o que realmente um MSX é capaz de fazer, mesmo sendo versão 1.1.

No campo profissional muitas novidades estão para acontecer e logo estará no mercado um software, desenvolvido para rodar no Expert DD Plus da Gradiente, que fará todo o controle de estoque, mala-direta e caixa de uma loja, utilizando caneta ótica, modem para envio/recepção de dados, etc.

Para que possamos fazer uma revista cada vez melhor, não deixe de nos enviar sua colaboração, sugestão ou mesmo crítica. Todas serão bem vindas e respondidas na medida do possível.

Um abraço,

Gonçalo Murteira

PAUTA

CASA DD MSX SE EXPANDE E INVESTE NA LINHA IBM-PC

A casa do MSX inaugurou duas lojas para usuários de micro computadores pessoais, uma em Curitiba, no Paraná e outra em São José do Rio Preto.

Com uma loja na capital e quatro no interior de São Paulo, a empresa tornou-se um importante redulo dos utilizadores de micros de 8 bits, mas decidiu investir, este ano, na linha dos "Qualid" de IBM-PC, fabricados pela Pointer, em Campinas-SP.

A Casa do MSX pretende tornar-se distribuidora exclusiva do PC Qualid gerações AT 286 e 386, mas não deixará de atender à demanda de seus consumidores de MSX e "DD Plus" da Gradiente.

O diretor da rede distribuidora, Renato Malato, não pretende alterar a "marca" da empresa, mas acredita que a interrupção no fornecimento de equipamentos da linha 8 bits pela Sharp e pela Gradiente, no ano passado, trouxe uma crise de desconfiança ao setor.

Malato está convencido de que consumidores de micros pessoais vêm se tornando mais técnicos e já não temem adquirir um equipamento mais sofisticado. "Isso já se observa mesmo entre o público jovem, que constitui 60% dos clientes da Casa do MSX", afirma o empresário.

KONAMI E OS NOVOS LANÇAMENTDS PARA MSX

A Konami, softhouse do Rio de Janeiro, está colocando no mercado dois novos produtos, inéditos no mercado.

O SCREEN 8 possibilita que o usuário tenha as melhores cenas do seu filme predileto em seu micro MSX e o Memory Mapper é um super cartucho que, entre outras possibilidades, pode ser utilizado como drive 'C' e permite a utilização de muitos programas ainda inéditos no Brasil.

Maiores informações poderão ser obtidas diretamente na Konami, que fica na Rua Sete de Setembro 92 sala 2210 – Rio de Janeiro – RJ – 20050.

TECLADO DE TECNOLOGIA CAPACITIVA

A Microset está lançando no mercado um teclado com chaves profissionais (4mm de curso), tecnologia capacitiva, 84 teclas e compatibilidade com PC XT.

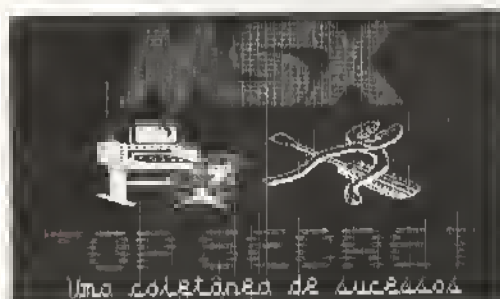
Uma de suas características é a gravação das teclas por processo de sublimação, o que impede que as legendas se desgastem com o uso.

O teclado oferece uma operação segura e confortável, gabinete low profile com pés ajustáveis e é apresentado em duas versões de teclas: feedeback tactil e a convencional.

Os preços, assim como o produto, são especiais e o telefone para maiores informações é (011) 263-8311.

AUTODESK DOA SOFTWARE PARA UNIVERSIDADES

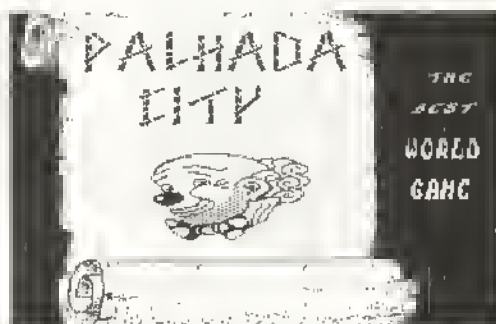
A Autodesk, produtora do best-seller mundial AutoCAD, anunciou o lança-

JUVENTUDE E TECNOLOGIA

Cadastramos Revendedores em Todo o Brasil

MSX TOP-SECRET

UM SUPER UTILITÁRIO TOTALMENTE INOVADOR NO MERCADO MSX, QUE COMPREENDE NADA MENOS QUE TODAS AS QUALIDADES DOS UTILITÁRIOS QUE JÁ FIZERAM SUCESSO NO MERCADO MSX. COM O TOP-SECRET É POSSÍVEL: ZAPPEAR UM DISQUETE, COPIAR E FORMATAR UM DISCO COM INCRÍVEL FIDELIDADE, EDITAR SPRITES 8X8 E 16X16, TRANSFORMAR E ELABORAR GRÁFICOS COM O COMANDO DRAW DO BASIC, ARMAZENAR SUAS INFORMAÇÕES EM UM SUPER BANCO DE DADOS, EDITAR ALFABETOS PARA USO EM QUALQUER SCREEN, IMPRIMIR TELAS E OUTRAS SUPER FERRAMENTAS. **Cr$ 1.100,00**

PALHADA CITY

PROVANDO QUE CRIATIVIDADE É DIFERENTE DE NOME COMERCIAL, A YOUNGSOFT LANÇA UM SENSACIONAL ADVENTURE QUE ALTERNA REALIDADE E FICÇÃO, DRAMA E COMÉDIA EM UMA MISTURA HOMOGÊNEA DE MUITO SUSPENSE E BOM HUMOR. RIO DE JANEIRO 01/01/2000. É NESSA DATA HISTÓRICA QUE ARIONALDINO, FAMOSO HISTORIADOR HOLANDÊS DESCOBRE QUE NA PACATA PALHADA CITY EXISTEM MUITOS SERES ALIENÍGENAS E ALIENADOS QUE ESCONDEM A MAIOR FORTUNA EM PEDRAS PRECIOSAS DO MUNDO. CONSEGUIRA ARIONALDINO ESCAPAR DE PALHADA CITY?
GRÁFICOS BELÍSSIMOS E ÓTIMA TRILHA SONORA. **Cr$ 550,00**

Cadastramos Revendedores em Todo Brasil

YOUNG CLUB APRENDIZADO E INFORMATIZAÇÃO

A FORMA MAIS OUSADA DE COMPARTILHAR CONHECIMENTOS JÁ É UMA REALIDADE.
CONFIRA OS BENEFÍCIOS:

- 50% de descontos em pacotes
- 30% de desconto em aplicativos de domínio público
- 10% de desconto em aplicativos nacionais
- 30% de desconto no aplicativo MSX Top Secret
- 20% de desconto no Adventure Palhada City
- 15% de desconto em compras da revista CPU
- Possibilidade de compartilhar idéias com outros sócios através da linha direta
- Young Express Disk uma soft-revista que lhe dará mensalmente dicas e truques de programação e macetes para jogos.

Obs.: O associado só terá direito aos benefícios após o segundo pagamento. Pagamento mensal.

MENSALIDADE: 10 BTN'S

COMO FAZER SEU PEDIDO:

Envie carta detalhada contendo os dados de seu equipamento e todas as informações sobre o seu pedido, anexando cheque nominal a Yvone di Santo Barboza no valor total de seu pedido.

PEÇA CATÁLOGO GRÁTIS

PACOTE DE JOGOS

Temos os mais variados tipos de pacotes a preços alucinantes.
Pacote de 05 e 10 jogos novidades Megaram
MSX Top Secret e Palhada City, estão registrados na associação das softhouses do Rio de Janeiro.
Lute contra a pirataria faça sua denúncia pelo tele-jovem.

Youngsoft
INFORMÁTICA LTDA.
A 1ª SOFTHOUSE DE NOVA IGUAÇU
Travessa Almerinda Lucas de Azeredo 11/1211 – Centro – Nova Iguaçu – Rio de Janeiro – CEP 26.210
(021) 767-9545 / Ramal 90

mento de um projeto educacional que beneficiará universidades brasileiras. Trata-se do "Education Grants Program", que contemplará as instituições com cópias do software AutoCAD para cada departamento técnico. A inscrição é simples: as universidades interessadas deverão preencher uma ficha, até o final de maio, que poderá ser obtida na Digicon, o distribuidor brasileiro dos produtos Autodesk.

O documento permitirá à empresa americana avaliar as condições técnicas para utilização do software. A universidade aprovada no programa pagará apenas as despesas alfandegárias e de transporte para receber sua cópia.

"A Autodesk já vinha colaborando com as instituições de ensino no Brasil, fornecendo softwares a preços subsidiados em 70%" – diz Gal Kimel, gerente para a América Latina e Israel. "No entanto, muitas universidades não dispunham de verba para adquirir o produto. Com o Education Grants Program, esses organismos poderão ter acesso a uma ferramenta que já conta com 300 mil usuários em todo o mundo".

Segundo Gal Kimel, que veio ao Brasil especialmente para o lançamento do programa, o objetivo da Autodesk é difundir a cultura AutoCAD, que pode auxiliar tecnologicamente o País, ajudando a incrementar sua produtividade. "O Brasil precisa ser competitivo lá fora e o AutoCAD é uma ferramenta que responde a essas necessidades", salienta ele. O AutoCAD roda em mais de 30 plataformas diferentes, de PCs a workstations, sob nove sistemas operacionais, entre eles o Unix, Xenix e DOS. No mundo todo já existem 700 aplicativos desenvolvidos a partir dele.

A intenção da Autodesk – que através de programas similares treina hoje em todo o mundo cerca de 100 mil pessoas – é estender gradativamente esses benefícios a órgãos estaduais e federais de ensino.

O endereço da Digicon, para a retirada do formulário, é Av. Paulista 726 3º andar sala 304 – São Paulo – SP. O telefone para esclarecimentos é (011) 285-41901.

UM ANO DE GARANTIA PARA A AMÉLIA 250

As impressoras desenvolvidas pela Elgin Eletrônica têm se destacado dentro do mercado nacional, devido à competência por elas apresentada. Possuindo moderna tecnologia, ótimos desempenho e rendimento, sobressaem-se substancialmente em relação às demais quando os assuntos são assistência técnica e prazos de garantia.

Prova disso é que a Elgin Eletrônica, além de estar instalando um centro próprio e exclusivo para a administração e manutenção de toda linha de impressoras, passa a oferecer um ano de garantia para a Amélia 250.

Enquanto o usual sempre foi facultar três meses de garantia aos produtos, a Amélia 250 já continha o maior prazo de garantia dentro de sua categoria, ou seja, quatro meses. Agora, este prazo foi triplicado e atinge 12 meses de garantia.

ESTABILIZADORES ESTÁTICOS PARA MICROCOMPUTADORES

A Televolt está colocando no mercado os estabilizadores Estáticos EE850 e EE1000, que foram especialmente desenvolvidos para fornecerem total proteção a microcomputadores, periféricos, telex e fac-símile, oferecendo uma série de vantagens.

Estes dois modelos combinam circuitos eletrônicos CI com transformadores para picos de até 850/1500 VA, garantindo uma tensão de saída constante e sem harmônicos na onda senoidal. São equipados com um excelente filtro de linha, protegendo o seu micro contra ruídos, estática e espúrias, que normalmente transmitam pela rede elétrica.

O endereço da Televolt, para maiores informações, é Rua dos Buritis 830 – São Paulo – SP – telefone (011) 275-2577.

REVOLUTION SOFTWARE

SUPERPROMOÇÕES PLANO BRASIL NOVO

- TUOO PARA EOUIPAR SEU MSX I – MSX II & MEGARAM
- TUOO CONGELADO COM PREÇO OE FEVEREIRO
- ATENOIMENTO P/OISKETTE OU FITA NA MESMA HORA
- ATENOIMENTO PELO CORREIO ULTRA RÁPIDO
- ÚLTIMAS NOVIOAOES VINOAS OO EXTERIOR P/MSX I & II
- TUOO EM PROMOÇÃO COM OESCONTOS OU BRINOE GRÁTIS
- FUNCIONAMOS OE SEG. A SAB., HORÁRIO COMERCIAL

PROMOÇÕES

1 – NA COMPRA DE QUALOUER EQUIPAMENTO ACIMA DE 10.000,00 CRUZEIROS GRÁTIS 15 GAMES OU APLICATIVO
2 – NA COMPRA DE 2 PACOTES GRÁTIS 5 APLICATIVOS OU FN 2 – 5 GAMES DEPENDENDO DO PACOTE COMPRADO
3 – NAS COMPRAS DE SOFTWARE ACIMA DE 5.000 CRUZEIROS 20% DE DESCONTO

TABELA DE PREÇOS

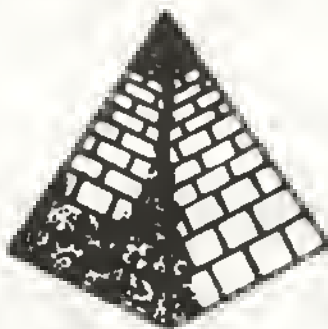
PACOTE COM 10 GAMES OU MEGAGAMES (1.1 2.0) .....
PACOTE COM 10 APLICATIVOS OU UTILITÁRIOS ....
PACOTE P/FITA K7 10 GAMES GAMES OU MEGAGAMES (UNIDADES) ....
DISKETTES NASHUA (10 UNIDADES) 5 1/4 ....

APLICATIVOS OU UTILITÁRIOS:

1 UNIDADE ....
2 UNIDADES ....
3 UNIDADES ....

LINHA DDX

DRIVE 5 1/4 360 Kb (COMPLETO)
DRIVE 5 1/4 720 Kb (COMPLETO)
DRIVE 3 1/2 720 Kb (COMPLETO)
TRANSFORMAÇÃO P/MSX 2.0 (24 HORAS)
MEGARAM DISK 256 Kb (SIMULA DRIVE)
EXPANSOR DE SLOTS (4 SLOTS C/FONTE)
PLACA 80 COLUNAS (DDX)
KIT DE DRIVE (GABINETE, FONTE, INTERF.)

REPRESENTANTE MSX SOFT INFORMÁTICA

AGORA SÓ NÃO COMPRA QUEM NÃO QUER

LINHA GRAOIENTE

EXPERT DD-PLUS C/DRIVE 3 1/2
EXPERT PLUS
MONITOR FÓSFORO VERDE
MODEM COM DISCAGEM AUTOMÁTICA
PLACA 80 COLUNAS (C/EDITOR DE TEXTOS)
GRAVADOR (DR-1)
JOYSTICK (ANALÓGICO)
PHANTON SYSTEM (NINTENDO)

LINHA TELCOM

MODEM D/MANUAL (MX-1)
MODEM D/AUTOM. (MX-2)
MODEM D/MANUAL (MX-3 VÍDEO TEXTO RES.)
MODEM D/AUTOM. (MX-4 VÍDEO TEXTO RES.
MODEM D/AUTOM. (MX-5 RESPOSTA AUTOM.)

LINHA TACTO

TRANSFORMAÇÃO TURBO (CLOCK 7,5 MHz)
EXPANSOR SLOTS (4 SLOTS)
CABO EXPANSOR
FONTE EXPANSOR (5 VOLTS)

OIVERSOS

IMPRESSORA ELGIN LADY 80
SUPORTE P/IMPRESSORA GD
SUPORTE P/IMPRESSORA PQ
FITA DE IMPRESSORA (LADY 80 & MTA)
CABO P/ IMPRESSORA (HOTBIT/PC)
CABO P/ IMPRESSORA (EXPERT)
FORMULÁRIO 80 COL. (1.000 FOLHAS)
ESTABILIZADOR TELEVOLT (EC-1 MANUAL)
ESTABILIZADOR TELEVOLT (EE-850 AUTOM.)

REVOLUTION SOFTWARE
AV. PRESIOENTE VARGAS 633 SALA 2120 CEP: 20071
CENTRO – RJ – PRÓXIMO METRÔ URUGUAIANA

ELGIN MOVIMENTA O MERCADO DO BRASIL NOVO

Passado um mês da decretação do Plano Brasil Novo, que num primeiro momento provocou uma paralisia completa no segmento de bens de informática, algumas empresas do setor, embora inseridas num mercado cada vez mais exigente e concorrido, aos poucos têm dado mostras de recuperação.

Nesse favorável contexto de retomada das negociações se destaca a Elgin Eletrônica, fabricante de impressoras desde 1982, que acaba de fechar contrato de OEM com a DFV Automação e Robótica para o fornecimento de impressoras MT 440 colorida.

Desenvolvida com a avançada tecnologia Elgin, a MT 440 colorida se destina às estações de trabalho montadas e comercializadas pela DFV, no sentido de imprimir telas gráficas que mostram situações instantâneas de controle de processo.

"A escolha pela MT 440 para compor a linha de funções da DFV se deu justamente pelas características e vantagens dessa impressora", afirma Reinaldo Marques Rangel, diretor de marketing da Elgin. Segundo Rangel, além da MT 440 ser a única impressora colorida (gera até sete cores) existente no mercado nacional, é também capaz de imprimir com velocidade de 400 cps textos e gráficos com alta definição e resolução.

CURSOS PARA MSX NO RIO

A Universal está lecionando cursos de Basic, dBase II e Wordstar para os micros da linha MSX.

Os cursos são práticos, onde o aluno aprende utilizando o equipamento, sendo uma boa opção para o pessoal do Rio de Janeiro que quer entrar no mundo da informática e desvendar os mistérios do seu MSX.

O endereço da Universal Computer é Rua Luis Guimarães 89 – Vila Isabel – telefone (021) 577-5356.

MEGA ASSEMBLER

Em 1987 a Cibertron Eletrônica lançou o software Mega Assembler de autoria de César Augusto O. Tiossi e José Alvaro Toledo Jr., que cederam os direitos de comercialização para a Cibertron, em troca de Royalts.

Este programa foi objeto de análise pela imprensa especializada, tendo sido comentado pela Folha de São Paulo de 23 de setembro de 1987 e 17 de fevereiro de 1988 e na revista Micro Sistemas de março de 1988.

No entanto, recentemente, uma empresa tem anunciado e vendido este software, utilizando, inclusive, o mesmo nome. Alertamos que esta prática é ilegal e que quem adquirir cópias que não sejam da Cibertron, possuirá um software pirata.

Os leitores que desejarem esclarecimentos, contactar a Cibertron, pelo telefone (011) 298-3299.

MULTIMODEM MSX DA TELCOM

O Multimodem MSX da Telcom vai tirar você e seu microcomputador MSX do isolamento, permitindo a interligação com outros microcomputadores (MSX, IBM-PC, APPLE, CP-500, etc.), com vários bancos de dados e sistemas, além de muitos outros CBBSs. Isto significa que você pode participar efetivamente da Comunidade de Teleinformática. O que você ganha com isso? O acesso a informações antes só disponíveis a quem possuía caros e sofisticados computadores. Ganha também a oportunidade de fazer novas amizades com pessoas que vivem o dia a dia da informática, e que certamente lhe proporcionarão oportunidades para trocar e obter novos conhecimentos. Se você é profissional liberal ou empresário, o Multimodem MSX irá abrir-lhe todo um novo horizonte para facilitar a sua vida profissional e ampliar os seus negócios.

BROKERS INICIA OPERAÇÕES COM SUPERCOMPUTADORES IBM 3090

Lançados no mercado há cerca de 4 anos, os supercomputadores IBM 3090 chegam agora ao chamado mercado secundário (de máquinas usadas), através da Brokers Consultores, Comércio e Representações Ltda.

Atuando na venda e agenciamento de operações com equipamentos usados IBM, a Brokers é empresa líder no mercado de secundários, tendo faturado US$ 22 milhões em 89.

Segundo seu Diretor, Necésio Tavares, o início das operações com o 3090 é uma decorrência natural do processo de desenvolvimento e hardware e da expansão das empresas, além do fato que, atualmente, já existem no Brasil cerca de 100 usuários do 3090, o que constitui uma satisfatória massa crítica. Até 1989, os usários do 3090 que sentiam necessidades de aumentar sua capacidade de processamento, chamavam a própria IBM para expansão do equipamento: "O que acontece é que os custos desta expansão não são compensadores", afirma Necésio Tavares, demonstrando na ponta do lápis as vantagens da venda da CPU para aquisição de um modelo de maior capacidade.

Por outro lado, os usuários dos sistemas 4341 e, principalmente, 4381, também estão começando a buscar uma solução maior, no caso, o 3090. E é aí que a Brokers entra em ação, oferecendo a linha 3090 a um custo inicial 20% inferior ao preço-lista da IBM, acrescidos de certificados de procedência e garantia, fornecidos pela Brokers e que incluem perícia, embalagem e transporte dos CPUs.

Segundo estimativa da própria IBM, os sistemas 3090 deverão ter em 1990 uma expansão de 40%, mas Necésio Tavares não tem dúvidas que pelo menos a metade deste total (ou seja, cerca de 20 usuários) poderão ser atendidos pela Brokers.

ANTI-PIRATA DA FARAH'S PROTEGE ATÉ CONTRA VÍRUS

Agora, no Brasil, o sistem "Disk-Lock", um "anti-pirata" consagrado nos EUA, é instalado pela Farah's Informática em qualquer PC. O sistema funciona a nível de hardware, não podendo, portanto, ser "enganado" ou anulado por nenhum programa ou usuário, por mais esperto que possam ser. Trata-se de uma chave eletrônica que somente permite a gravação de dados em disquete se a chave-mestra estiver na posição "Cópia autorizada".

O sistema, garante Tanios Hamzo, engenheiro da Farah's, impede que informações contidas em "winchesters" sejam duplicadas pelo "disk drive" sem permissão prévia. O sistema é tão seguro que, mesmo que um especialista desmonte o computador, assim mesmo levará horas até desativar o Disk-Lock. Pode-se enganar programas, mas não se pode enganar a eletrônica.

Segredos industriais, financeiros e outras informações sigilosas não correrão mais o risco de "vazar" para terceiros, pois uma simples cláusula acrescida a qualquer programa pode testar rápida e facilmente se há autorização para cópia pelo "Disk-Lock".

Mas talvez o melhor de tudo isso seja a proteção intrínseca que o sistema oferece contra os chamados "vírus" de computador, uma vez que nenhum dado pode ser gravado ou adulterado sem autorização.

Para maiores detalhes, o telefone da Farah's é (011) 37-3437.

POR DENTRO DA INTERFACE DE DRIVE

PARTE II

Júlio Velloso

Dando prosseguimento ao artigo anterior, apresentarei, neste número, as rotinas mencionadas anteriormente.

Achei interessante debugar parte da interface de drive, afim de termos informações importantes, que podem ser usadas em programas baseados tanto nas rotinas aqui publicadas, como em outras desenvolvidas por outros profissionais do mercado.

Estas rotinas lazem parte das rotinas básicas do programa 'Sistema Operacional BKP DISCO', podendo ser usadas por qualquer pessoa. O restante das rotinas ainda não publicadas não estão disponíveis ao público, estando, portanto, sujeito a processo o individuo que fizer uso, copiar parte ou inteiramente o programa.

A TABELA DE LOCAÇÃO DOS AROUIVDS – FAT (FILE ALOCATION TABLE) –

Como foi dito no artigo anterior de CPU, é esta parte do disco que contém as informações a respeito de que áreas do disco estão ou não disponíveis ao usuário.

Estas informações estão guardadas seguindo um processo que em informática recebe o nome de 'estrutura relacional'.

Em programas profissionais, o programador encontra sempre dificutdades na otimização da velocidade de arquivamento de dados.

Solução para este problema é o arquivamento da estrutura dita anteriormente, que guarda estes dados em ordem da sua chave. Nesle caso, se você desejar acessar o dado que possui a chave 4, você deve acessar o quarto dado. Teríamos:

chave	dado.
001	dado1
003	dado2
005	dado3
008	dado4

A chave de acesso pode representar qualquer elemento, como data, número de chamada de uma turma de colégio elc.

Esta tabeta na memória ficaria da seguinte forma:

end01	=	dado1
end05	=	vazio
end09	=	dado2
end13	=	vazio
end17	=	dado3
end21	=	vazio
end25	=	vazio
end29	=	dado4

Supondo que para cada registro tem-se 4 bytes, se quizermos acessar o registro da chave 5, teríamos que fazer a operação (5-1)*4+1), que resultaria em 17. Generalizando, se quizermos acessar o registro de cnave N, faríamos ((N-1)*4+1). Verifique se dá cerlo?

É evidente que esta forma de acesso a dados é boa se tivermos chaves baixas. Do contrário, se tivermos chaves de número alto, como 16000, teríamos que ler uma memória de 16000*TAMREG bytes, onde TAMREG é o tamanho do registro em bytes.

No caso do exemplo acima, precisaríamos de 80 Kbytes, e, se tivermos apenas 5 dados, teríamos quase que 79975 bytes não utilizados, o que seria inviável.

No exemplo da FAT, o computador faz exatamente este tipo de arquivamento, sendo que os dados são os seguintes:

000 – cluster apagado ou vazio
FFF – final de arquivo ou cluster não disponível.
NNN – próximo cluster da labela.

Em um exemplo prálico, se um arquivo ocupa os seguintes clusters de um disco:

002 003 094 008 009 00A

se tormaria a seguinle tabeta:

chave (cluster)	dado
002	003
003	004
004	008
005	000
006	000
007	000
008	009
009	010
010	FFF

que, na memória, ficaria da seguinte forma:

END00 03 00 04 00 08 00 00 00
END08 00 00 00 00 00 09 00 0A
END10 FF OF

ECDNOMIZANDD BYTES

O aulor do DOS constatou que um cluster ocupa, no máximo, 1 byte e meio, e que em 3 bytes ele poderia colocar 2 clusters, e economizar bytes.

O padrão adotado por ele, para compactar estes clusters, é o seguinte:

byte 1: x1/x2
byte 2: x3/x4
byte 3: x5/x6

OBS.: cada x representa metade de um byte.

1º cluste: x4/x1/x2
2º cluster: x5/x6/x3

No exemplo anterior, a memória ficaria da seguinte forma:

END00 03 40 00 08 00 00 00 00
END08 00 09 A0 00 FF 0F 00 00
END10 00

Nolem que, de 17 bytes, este recurso economizou 3 bytes.

Voltando ao nosso exemplo, outras informações que teriam que ser registradas são: o tipo de formatação, a entrada e o lamanho real do arquivo.

Na FAT só a formatação é registrada através de um byte, (a relação do byte com o tipo de formatação foi descrita no artigo anterior).

No caso de discos de face dupla, 40 trilhas, o byte é FDh. A FAT, em nosso exemplo, licaria da seguinte forma:

END00 FD FF FF 03 40 00 08 00
END08 00 00 00 00 09 A0 00 FF
END10 0F 00 00 00

Para disco de 40 trilhas face simples ou dupla, e 80 trilhas face simples, o número de clusters é de 354, que multipli-

cando por 2, ficaria 708 bytes, com mais 3 da formatação teremos 711, o que em hexadecimal seria igual a 2C7H. Compactando, ficaria, então, 354 * 3/2 = 535 com mais 3 da formatação 538, que em hexadecimal seria 216H. Precisaria, então, 2 setores, nos dois modos.

No exemplo acima, a economia passa de 3 bytes com 8 clusters para 177 bytes com 354 clusters.

Para discos de face dupla 80 trilhas, o número de clusters passa para 717. Fazendo a conta igual ao exemplo anterior, seria um total de 59AH bytes para arquivos descompactados e 433H bytes na forma compactada, o que daria, nos dois modos, 3 setores.

No exemplo acima, a economia ficou em 359 bytes para 717 clusters.

Como vimos nos exemplos anteriores, o modo compactado não economiza, em termos de setores, em relação ao não compactado. Apenas quando o número de cluster passar de 1440 é que o modo compactado passa a dar algum resultado.

Apenas os discos rígidos, os chamados 'Winchester's', possuem capacidade de guardar mais de 1440 clusters. É somente para estes casos que o modo compactado possui utilidade.

MEXENDO NA FAT

No caso do arquivo de 8 clusters, identificar e traduzir uma FAT se torna relativamente fácil, mas quando isto aumenta de complexidade e tamanho, passa a ser penoso identificar um arquivo nesta FAT.

Apresento uma tabela que relaciona endereço com cluster, para auxiliar o trabalho de tradução da FAT:

O número colocado ao lado do endereço se refere ao setor do disco onde se localiza o endereço.

Cada endereço acima é um dos segmentos internos da FAT com 2 partes.

Cada parte é um cluster do disco. De modo simplificado, um cluster agrupa o lado A e o lado B de um mesmo setor ou posição do disco. No caso do disco ser de 1 lado apenas, temos somente 1 setor no cluster.

SIMULANDO O BDOS

É muito complicado e penoso debugar todo o BDOS para aproveitá-lo e fazer uma rotina mais compacta e eficiente.

Neste artigo, mostrarei 9 rotinas que, em conjunto, possibilitarão fazer 5 opera-

ENDEREÇO	CLUSTER	ENDEREÇO	CLUSTER	ENDEREÇO	CLUSTER
1 003 ==	002	2 001 ==	156	3 002 ++	2AE
1 006 ==	004	2 004 ==	158	3 005 ++	2B0
1 009 ==	006	2 007 ==	15A	3 008 ++	2B2
1 00C ==	008	2 00A ==	15C	3 00B ++	2B4
1 00F ==	00A	2 00D ==	15E	3 00E ++	2B6
1 012 ==	00C	2 010 ==	160	3 011 ++	2B8
1 015 ==	00E	2 013 ==	162	3 014 ++	2BC
1 018 ==	010	2 016 ==	164	3 017 ++	2BE
1 01B ==	012	2 019 ++	166	3 01A ++	2C0
1 01E ==	014	2 01C ++	168	3 01D ++	2C2
1 021 ==	016	2 01F ++	16A	3 020 ++	2C4
1 024 ==	018	2 022 ++	16C	3 023 ++	2C6
1 027 ==	01A	2 025 ++	16E	3 026 ++	2C8
1 02A ==	01C	2 028 ++	170	3 029 ++	2CC
1 02D ==	01E	2 02B ++	172	3 02C ++	2CE
1 030 ==	020	2 02E ++	174	3 02F ++	2D0
1 033 ==	022	2 031 ++	176		
1 036 ==	024	2 034 ++	178		
1 039 ==	026	2 037 ++	17A		
1 03C ==	028	2 03A ++	17C		
1 03F ==	02A	2 03D ++	17E		
1 042 ==	02C	2 040 ++	180		
1 045 ==	02E	2 043 ++	182		
1 048 ==	030	2 046 ++	184		
1 04B ==	032	2 049 ++	186		
1 04E ==	034	2 04C ++	188		
1 051 ==	036	2 04F ++	18A		
1 054 ==	038	2 052 ++	18C		
1 057 ==	03A	2 055 ++	18E		
1 05A ==	03C	2 058 ++	190		
1 05D ==	03E	2 05B ++	192		
1 060 ==	040	2 05E ++	194		
1 063 ==	042	2 061 ++	196		
1 066 ==	044	2 064 ++	198		
1 069 ==	046	2 067 ++	19A		
1 06C ==	048	2 06A ++	19C		
1 06F ==	04A	2 06D ++	19E		
1 072 ==	04C	2 070 ++	1A0		
1 075 ==	04E	2 073 ++	1A2		
1 078 ==	050	2 076 ++	1A4		
1 07B ==	052	2 079 ++	1A6		
1 07E ==	054	2 07C ++	1A8		
1 081 ==	056	2 07F ++	1AA		
1 084 ==	058	2 082 ++	1AC		
1 087 ==	05A	2 085 ++	1AE		
1 08A ==	05C	3 088 ++	1B0		
1 08D ==	05E	2 08B ++	1B2		
1 090 ==	060	2 08E ++	1B4		
1 093 ==	062	2 091 ++	1B6		
1 096 ==	064	2 094 ++	1B8		
1 099 ==	066	2 097 ++	1BA		
1 09C ==	068	2 09A ++	1BC		
1 09F ==	06A	2 09D ++	1BE		
1 0A2 ==	06C	2 0A0 ++	1C0		
1 0A5 ==	06E	2 0A3 ++	1C2		
1 0A8 ==	070	2 0A6 ++	1C4		
1 0AB ==	072	2 0A9 ++	1C6		
1 0AE ==	074	2 0AC ++	1C8		
1 0B1 ==	076	2 0AF ++	1CA		
1 0B4 ==	078	2 0B2 ++	1CC		
1 0B7 ==	07A	2 0B5 ++	1CE		
1 0BA ==	07C	2 0B8 ++	1D0		
1 0BD ==	07E	2 0BB ++	1D2		
1 0B0 ==	080	2 0BE ++	1D4		
1 0C3 ==	082	2 0C1 ++	1D6		
1 0C6 ==	084	2 0C4 ++	1D8		
1 0C9 ==	086	2 0C7 ++	1DA		
1 0CC ==	088	2 0CA ++	1DC		
1 0CF ==	08A	2 0CD ++	1DE		
1 0D2 ==	08C	2 0D0 ++	1E0		
1 0D5 ==	08E	2 0D3 ++	1E2		
1 0D8 ==	090	2 0D6 ++	1E4		
1 0DB ==	092	2 0D9 ++	1E6		
1 0DE ==	094	2 0DC ++	1E8		
1 0E1 ==	096	2 0DF ++	1EA		
1 0E4 ==	098	2 0E2 ++	1EC		
1 0E7 ==	09A	2 0E5 ++	1EE		
1 0EA ==	09C	2 0E8 ++	1F0		
1 0ED ==	09E	2 0EB ++	1F2		
1 0F0 ==	0A0	2 0EE ++	1F4		
1 0F3 ==	0A2	2 0F1 ++	1F6		

== significa que tanto a 1ª parte quanto a 2ª parte do segmento é usada. Todas as formatações.

++ significa que tanto a 1ª parte quanto a 2ª parte do segmento é usada. Só para discos de 80 trilhas face dupla.

ENDEREÇO	CLUSTER	ENDEREÇO	CLUSTER
1 0F6 ==	0A4	2 0F4 ++	1F8
1 0F9 ==	0A6	2 0F7 ++	1FA
1 0FC ==	0A8	2 0FA ++	1FC
1 0FF ==	0AA	2 0FD ++	1FE
1 102 ==	0AC	2 100 ++	200
1 105 ==	0AE	2 103 ++	202
1 108 ==	0B0	2 106 ++	204
1 10B ==	0B2	2 109 ++	206
1 10E ==	0B4	2 10C ++	208
1 111 ==	0B6	2 10F ++	20A
1 114 ==	0B8	2 112 ++	20C
1 117 ==	0BA	2 115 ++	20E
1 11A ==	0BC	2 118 ++	210
1 11D ==	0BE	2 11B ++	212
1 120 ==	0C0	2 11E ++	214
1 123 ==	0C2	2 121 ++	216
1 162 ==	0C4	2 124 ++	218
1 129 ==	0C6	2 127 ++	21A
1 12C ==	0C8	2 12A ++	21C
1 12F ==	0CA	2 12D ++	21E
1 132 ==	0CC	2 130 ++	220
1 135 ==	0CE	2 133 ++	222
1 138 ==	0D0	2 136 ++	224
1 12B ==	0D2	2 139 ++	226
1 13E ==	0D4	2 13C ++	228
1 141 ==	0D6	2 13F ++	22A
1 144 ==	0D8	2 142 ++	22C
1 147 ==	0DA	2 145 ++	22E
1 14A ==	0DC	2 148 ++	230
1 14D ==	0DE	2 14B ++	232
1 150 ==	0E0	2 14E ++	234
1 153 ==	0E2	2 151 ++	236
1 156 ==	0E4	2 154 ++	238
1 159 ==	0E6	2 157 ++	23A
1 15C ==	0E8	2 15A ++	23C
1 15F ==	0EA	2 15D ++	23E
1 162 ==	0EC	2 160 ++	240
1 165 ==	0EE	2 163 ++	242
1 168 ==	0F0	2 166 ++	244
1 16B ==	0F2	2 169 ++	246
1 16E ==	0F4	2 16C ++	248
1 171 ==	0F6	2 16F ++	24A
1 174 ==	0F8	2 172 ++	24C
1 177 ==	0FA	2 175 ++	24E
1 17A ==	0FC	2 178 ++	250

ENDEREÇO	CLUSTER	ENDEREÇO	CLUSTER
1 17D ==	0FE	2 17B ++	252
1 180 ==	100	2 17E ++	254
1 183 ==	102	2 181 ++	256
1 186 ==	104	2 184 ++	258
1 189 ==	106	2 187 ++	25A
1 18C ==	108	2 18A ++	25C
1 18F ==	10A	2 18D ++	25E
1 192 ==	10C	2 190 ++	260
1 195 ==	10E	2 193 ++	262
1 198 ==	110	2 196 ++	264
1 19B ==	112	2 199 ++	266
1 19E ==	114	2 19C ++	268
1 1A1 ==	116	2 19F ++	26A
1 1A4 ==	118	2 1A2 ++	26C
1 1A7 ==	11A	2 1A5 ++	26E
1 1AA ==	11C	2 1A8 ++	270
1 1AD ==	11E	2 1AB ++	272
1 1B0 ==	120	2 1AE ++	274
1 1B3 ==	122	2 1B1 ++	276
1 1B6 ==	124	2 1B4 ++	278
1 1B9 ==	126	2 1B7 ++	27A
1 1BC ==	128	2 1BA ++	27C
1 1BF ==	12A	2 1BD ++	27E
1 1C2 ==	12C	2 1C0 ++	280
1 1C5 ==	12E	2 1C3 ++	282
1 1C8 ==	130	2 1C6 ++	284
1 1CB ==	132	2 1C9 ++	286
1 1CE ==	134	2 1CC ++	288
1 1D1 ==	136	2 1CF ++	28A
1 1D4 ==	138	2 1D2 ++	28C
1 1D7 ==	13A	2 1D5 ++	28E
1 1DA ==	13C	2 1D8 ++	290
1 1DD ==	13E	2 1DB ++	292
1 1E0 ==	140	2 1DE ++	294
1 1E3 ==	142	2 1E1 ++	296
1 1E6 ==	144	2 1E4 ++	298
1 1E9 ==	146	2 1E7 ++	29A
1 1EC ==	148	2 1EA ++	29C
1 1EF ==	14A	2 1ED ++	29E
1 1F2 ==	14C	2 1F0 ++	2A2
1 1F5 ==	14E	2 1F3 ++	2A4
1 1F8 ==	150	2 1F6 ++	2A6
1 1FB ==	152	2 1F9 ++	2A8
1 1FE ==	154	2 1FC ++	2AA
		2 1FF ++	2AC

ções que serão muito importantes para trabalhar com o disco:

As Rotinas:

STCONV – trabalha em conjunto com SQCONV. Ela simplesmente pega um cluster na FAT.

Entrada: IY – posição do segmento.
A – parte do segmento.
Saída: HL – cluster

TRQDSQ – inicializa variável interna
Entrada: A – drive.

SQCONV – monta uma tabela mais legível da FAT, separando-a por programas.
Saída: A – FF = FAT vazia
00 – ok

MNTDIR – monta tabela relacionando entrada do arquivo com o seu tamanho.

ERRMAP – parte de SQCONV é usado caso a FAT se encontre vazia.

CACFAT – calcula quanto, em bytes, representa A setores. Faz parte das rotinas SQCONV e ATLMAP.

Entrada: A – número de setores
Saída: BC – tamanho em bytes

ATLMAP – monta FAT baseado em tabela montada por SQCONV.

SAVFAT – salva FAT em disco, depois de alterada.

Entrada: A – FF = não faz BACKUP
00 = faz BACKUP

GRVNUM – coloca um número na FAT. Faz parte da rotina ATLMAP.

Entrada: A – parte do segmento
HL – cluster a ser gravado
IX – endereço do segmento

As Tabelas:

BUFPGM – é dividida por programas. Os clusters são inseridos de forma descompactada, de modo a facilitar o usuário.

NNN1	NNN2	NNN3	FFF	–	PGM1
NNN4	NNN5	NNN6	FFF	–	PGM2
"	"	"	"	–	PGM3
"	"	"_	"	–	PGM4
"	"	"	"	–	PGM5
NNNA	NNNB	NNNC	FFF	–	PGM6
FFFF				–	Final do Buffer

NNN – é uma word que possui 2 bytes. Representa um cluster.
OFFF – assinala final de arquivo.
FFFF – assinala final do buffer

BUFDIR – relaciona entrada com tamanho.

EEE1	TTT1	–	PGM1
EEE2	TTT2	–	PGM2
"	"		
EEE8	TTT8	–	PGM8
FFFF		–	Final da tabela

Programa Exemplo:

```
LD A,0
CALL TRQDSC
CALL SQCONV
CALL MNTDIR
RET
```

Na memória, irá aparecer a tradução da FAT com o conteúdo explicado acima e a relação da entrada com o tamanho.

O programa utiliza o drive A, pois no registrador A existe o valor zero, caso se queira trocar de drive, basta mudar este valor, utilizando (1 para o drive B, 2 para o drive C etc.).

O PROGRAMA EXAFAT

O programa da Listagem 2 coloca em prática as rotinas acima, fazendo uma análise de cada arquivo, mostrando os seus cluster's, b tamanho de cada arquivo e sua data de criação.

Para digitar este programa entre com a Listagem 1 usando o MSXDEBUG e gravando depois com o comando BSAVE. Depois, entre com a listagem 2, voltando ao BASIC.

O funcionamento do programa é apresentado na parte inferior da tela.

Nos próximos números de CPU, apresentarei as rotinas de carregamento e gravação de arquivo baseado nas rotinas apresentadas aqui. Listarei, também, as variáveis do sistema, que julguei serem úteis aos usuários. Aguardem. . .

Listagem 1

```
9000 C3 2C 90 AF C3 74 90 C3 13 91 C3 8B 92 C3 68 92
9010 C3 F7 92 C3 04 93 C3 7B 93 C3 C9 93 1B A2 1B A0
9020 1B A8 1B 94 1B 9A 7C 92 C0 7D 93 C9 FE 01 28 2E
9030 FD 7E 02 FD 36 02 00 F5 FD 7E 01 F5 E6 0F FD 77
9040 01 F1 E6 F0 0F 0F 0F 0F 6F F1 F5 E6 0F 07 07 07
9050 07 47 7D B0 6F F1 E6 F0 0F 0F 0F 0F 67 C9 FD 7E
9060 00 FD 36 00 00 6F FD 7E 01 F5 E6 F0 FD 77 01 F1
9070 E6 0F 67 C9 21 1B A2 11 01 00 01 FC 01 32 07 94
9080 A7 CD A7 FF 3A 1B A2 FE FC CC C7 90 FE FD CC DA
9090 90 FE F8 CC ED 90 FE F9 CC 00 91 22 FF 93 26 00
90A0 69 22 03 94 68 22 05 94 ED 53 08 94 FD E5 C1 78
90B0 32 0D 94 79 32 0A 94 DD E5 C1 69 22 08 94 3A FF
90C0 93 4F ED 43 01 94 C9 21 FC 01 01 09 07 11 00 02
90D0 FD 21 02 28 DD 21 05 04 AF C9 21 FD 02 01 0C 04
90E0 11 00 04 FD 21 02 28 DD 21 05 07 AF C9 21 F8 01
90F0 01 0C 0A 11 00 02 FD 21 02 50 DD 21 05 07 AF C9
9100 21 F9 02 01 0E 05 11 00 04 FD 21 03 50 DD 21 07
9110 07 AF C9 21 1B 94 11 01 00 ED 4B FF 93 3A 0A 94
9120 47 3A 07 94 A7 CD A7 FF 21 FF 0F 22 1B A2 21 1D
9130 A2 22 19 94 FD 21 1E 94 21 02 00 22 0F 94 21 00
9140 00 22 12 94 3E 01 32 11 94 3A 0A 94 CD F7 92 21
9150 1B 94 09 2B 7E A7 28 FB 23 22 17 94 11 1E A0 CD
9160 26 90 CA 68 92 FD 66 01 FD 6E 00 E5 3A 11 94 CD
9170 2C 90 7C B5 E1 FD 74 01 FD 75 00 20 03 CD 57 92
9180 2A 0F 94 CD 95 91 CD 57 92 11 FF 0F CD 26 90 CA
9190 A8 91 C3 CD 91 DD 2A 19 94 DD 75 00 DD 74 01 DD
91A0 23 DD 23 DD 22 19 94 C9 21 FF 0F CD 95 91 2A 12
91B0 94 7C B5 C4 06 92 CD 1D 92 CD 57 92 E5 2A 0F 94
91C0 CD 95 91 E1 11 FF 0F CD 26 90 CA A8 91 ED 5B 0F
91D0 94 A7 ED 52 E5 2B 7D B4 E1 C2 E2 91 CD 1D 92 C3
91E0 80 91 E5 2A 12 94 7D B4 20 10 2A 0F 94 22 12 94
91F0 3A 11 94 32 14 94 FD 22 15 94 C1 CD 1D 92 0B 79
9200 B0 20 F8 C3 B0 91 2A 12 94 22 0F 94 21 00 00 22
9210 12 94 3A 14 94 32 11 94 FD 2A 15 94 C9 E5 2A 0F
9220 94 23 22 0F 94 E1 3A 11 94 FE 02 28 06 3E 02 32
9230 11 94 C9 FD 23 FD 23 FD 23 3E 01 32 11 94 2A 12
9240 94 7C B5 C0 FD E5 E1 ED 5B 17 94 CD 26 90 D8 21
9250 FF FF CD 95 91 E1 C9 3A 11 94 CD 2C 90 7C B5 F5
9260 CC 1D 92 F1 28 F1 AF C9 2A 05 94 23 23 11 04 00
9270 A7 ED 52 22 1D A2 22 1B A0 21 00 02 22 1D A0 21
9280 FF FF 22 1F A2 22 1F A0 3E FF C9 21 1B A8 ED 5B
9290 0B 94 ED 4B 01 94 3A 07 94 A7 CD A7 FF DD 21 1B
92A0 A0 FD 21 1D A2 FD 66 01 FD 6E 00 DD 74 01 DD 75
92B0 00 21 00 00 FD 23 FD 23 ED 5B 08 94 19 E5 FD 7E
92C0 01 FD 6E 00 E6 0F 67 11 FF 0F CD 26 90 E1 20 E4
92D0 FD 23 FD 23 DD 74 03 DD 75 02 DD 23 DD 23 DD 23
92E0 DD 23 FD 66 01 FD 6E 00 11 FF FF CD 26 90 20 B8
92F0 DD 74 01 DD 75 00 C9 21 00 00 11 00 02 47 19 10
9300 FD E5 C1 C9 00 3A 0A 94 CD F7 92 FD 21 1D A2 DD
9310 21 1B 94 DD E5 DD E5 E1 D1 13 36 00 ED B0 3A FF
9320 93 DD 77 00 DD 36 01 FF DD 36 02 FF FD 66 01 FD
9330 6E 00 44 4D DD 21 1E 94 0B 0B 78 B1 3E 00 28 13
9340 A7 28 08 DD 23 DD 23 DD 23 3E FF 3C 0B 57 78 B1
9350 7A 20 ED FD 23 FD 23 FD 66 01 FD 6E 00 CD C9 93
9360 11 FF 0F CD 26 90 20 CA FD 23 FD 23 FD 66 01 FD
9370 6E 00 11 FF FF CD 26 90 20 B8 C9 00 FE FF 28 43
9380 21 1B 9A E5 11 01 00 ED 4B FF 93 3A 0A 94 47 3A
9390 07 94 A7 CD A7 FF E1 3A 0A 94 3C 5F 16 00 ED 4B
93A0 FF 93 3A 0A 94 47 3A 07 94 37 CD A7 FF 21 1B 94
93B0 11 01 00 ED 4B FF 93 3A 0A 94 47 3A 07 94 37 CD
93C0 A7 FF C9 21 1B 94 E5 18 CD FE 01 28 0F DD 75 00
93D0 7C E6 0F 4F DD 7E 01 B1 DD 77 01 C9 7D E6 F0 0F
93E0 0F 0F 0F 4F 7C E6 0F 07 07 07 07 B1 DD 77 02 7D
93F0 E6 0F 07 07 07 07 4F DD 7E 01 B1 DD 77 01 C9 00
9400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Listagem 2

```
10 '
20 ' Programa Exemplo Para Teste de EXAA
RQ
30 ' Por: Júlio Renato Soares Velloso
40 '
100 SCREEN0:KEYOFF:DEFINTA-Z:POKE&HF351,
&H0:POKE&HF352,&H90
110 PRINT
120 INPUT "Qual o setor desejado";S
130 PRINTDSKI$(1,S)
140 S$="00"
150 E=&H9000
160 FOR R=1 TO 1:N=E
170 PRINT HEX$(N);" ";
180 FOR I=1 TO 16
190 B$=HEX$(PEEK(N)):T=2-(LEN(B$))
200 PRINT MID$(S$,1,T);B$;" ";
210 N=N+1
```

```
220 NEXT I
230 N=E:FOR I=1 TO 16
240 IF PEEK(N)>31 THEN PRINT CHR$(PEEK(N
)); ELSE PRINT".";
250 N=N+1
260 NEXT I
270 PRINT
280 IF E>&H91E0 THEN A$=INPUT$(1):GOTO 1
10
290 NEXTR:E=N:GOTO160
1000 '
1010 '
1020 '        Programa: EXEFAT (EXAminado
r da FAT)
1030 '        Direitos: Revista CPU da Ag
uia Informática LTDA
1040 '              Por: Júlio Renato Soar
es Velloso
1050 '
1060 '
1070 SCREEN0:WIDTH80:COLOR,1,1:KEYOFF:CL
EAR1200,&HA000:DEFINTA-Z:LOCATE,,0:GOSUB
1650
1080 '
1090 BLOAD"DRIVE2.BIN"
1100 '
1110 DEFUSR1=&H9000:DEFUSR2=&H9003:DEFUS
R3=&H9007:DEFUSR4=&H900A:DEFUSR6=&H900D:
DEFUSR7=&H9010:DEFUSR8=&H9013:DEFUSR9=&H
9016:DEFUSR0=&H9019
1120 '
1130 A=USR2(0):A=USR3(0):A=USR4(0):A=USR
7(0):GOSUB1760
1140
1150 BPGM!=256*PEEK(&H901D)+PEEK(&H901C)
:BDIR!=256*PEEK(&H901F)+PEEK(&H901E):DIR
!=256*PEEK(&H9021)+PEEK(&H9020):FAT!=256
*PEEK(&H9023)+PEEK(&H9022):BFAT'=256*PEE
K(&H9025)+PEEK(&H9024)
1160 '
1170 GOSUB 1920
1180 '
1190 PGN!=BPGM!+2
1200 '
1210 ADIR!=DIR!:SE!=256*PEEK(PGM!+1)+PEE
K(PGN!):IF SE!=65535! GOTO 1600
1220 CI!=PEEK(ADIR!+27)*256+PEEK(ADIR!+2
6)
1230 IF CI!<>SE! THEN ADIR!=ADIR!+32:IF
PEEK(ADIR!)<>0 GOTO 1220
1240 '
1250 LOCATE16,6:FORI=0TO7:PRINTCHR$(PEEK
(ADIR!+I));:NEXTI:PRINT".";:FORI=8TO10:P
RINTCHR$(PEEK(ADIR!+I));:NEXTI
1260 '
1270 LOCATE44,6:PRINTPEEK(ADIR!+29)*256+
PEEK(ADIR!+28)+PEEK(ADIR!+31)*4096+PEEK(
ADIR!+30)*65536!;"    ";
1280 '
1290 LOCATE64,6:DAT!=PEEK(ADIR!+25)*256+
PEEK(ADIR!+24):PRINT(DAT! AND &H1F);"/";
(DAT! AND &H1E0)/32;"/";((DAT! AND &HFE0
0)/256)+74
1300 '
1310 NM=1:NY=8:LOCATE 15,NY
1320 SE=256*PEEK(PGN!+1)+PEEK(PGN!):IF S
E=&HFFF GOTO 1370
1330 PRINT (SE+PEEK(&H9405))*2;"  ";:PGN
!=PGM!+2
1340 NN=NN+1:IF NM=9 THEN NN=1:NY=NY+1:L
OCATE15,NY
1350 GOTO 1320
1360 '
1370 A$=INKEY$:IF A$="" GOTO 1370
1380 IF ASC(A$)=28 THEN PGN!=PGM!+2:GOSU
B 1470:GOTO 1210
1390 IF ASC(A$)=29 GOTO 1530
1400 IF ASC(A$)=30 THEN GOSUB 1470:GOTO
1190
1410 IF ASC(A$)=31 GOTO 1580
1420 IF ASC(A$)=27 THEN CLS:KEY ON:END
1430 IF ASC(A$)=12 THEN GOSUB1470:GOTO 1
100
1440 IF ASC(A$)=24 GOTO 1810
1450 GOTO 1370
1460 '
1470 POKE&HBA01,5:POKE&HBA08,52:LOCATE16
,8:A=USR(0):RETURN
1480 '
1490 POKE&HBA01,14:POKE&HBA08,77:LOCATE1
,5:A=USR(0):RETURN
1500 '
1510 POKE&HBA01,3:POKE&HBA00,76:LOCATE2,
20:A=USR(0):RETURN
1520 '
1530 GOSUB 1620
1540 IF PGM!=BPGM! GOTO 1560
1550 GOSUB 1620
1560 PGN!=PGN!+2:GOSUB 1470:GOTO 1210
1570 '
1580 GOSUB 1470
1590 PGN!=PGM!+2:SE!=PEEK(PGN!+1)*256+PE
EK(PGM!):IF SE!<>65535! GOTO 1590
1600 PGM!=PGN!-2:GOSUB 1620:PGN!=PGN!+2:
GOTO 1210
1610 '
1620 PGN!=PGM!-2:SE!=PEEK(PGM!+1)*256+PE
EK(PGM!):IF SE!<>4095 GOTO 1620
1630 RETURN
1640 '
1650 DEFUSR=&H41:A=USR(0):T1$=CHR$(1)+CH
R$(64+23):T2$=CHR$(1)+CHR$(64+22):T3$=CH
R$(1)+CHR$(64+24):T4$=CHR$(1)+CHR$(64+25
):T5$=CHR$(1)+CHR$(64+26):T6$=CHR$(1)+CH
R$(64+27):T7$=CHR$(1)+CHR$(64+20):T8$=CH
R$(1)+CHR$(64+19)
1660 LOCATE2,1:PRINT"Programa: EXEFAT (E
xaminador da FAT)"
1670 LOCATE5,3:PRINT"Autor: Júlio Renato
 Soares Velloso";SPC(17);"(C) 1990"
1680 LOCATE69,1:PRINTCHR$(&HC7)CHR$(&HD3
)" "CHR$(&HC7)CHR$(&HC1)" "CHR$(&HDD)CHR
$(&HDE)
1690 LOCATE69,2:PRINTCHR$(&HDD)"  "CHR$(
&HC1)CHR$(&HC7)CHR$(&H20)CHR$(&HDD)CHR$(
&HDE)
1700 LOCATE69,3:PRINTCHR$(&HC1)CHR$(&HD6
)" "CHR$(&HDD)"  "CHR$(&HC1)CHR$(&HC7)
1710 GOSUB 1950
1720 FORI=0TO78:LOCATEI,4:PRINTT1$;:LOCA
TEI,0:PRINTT1$;:LOCATEI,19:PRINTT1$;:LOC
ATEI,24:PRINTT1$;:IFI<25THENLOCATE0,I:PR
INTT2$;:LOCATE78,I:PRINTT2$;:NEXTIELSENE
XTI
1730 LOCATE0,0:PRINTT3$;:LOCATE78,0:PRIN
TT4$;:LOCATE0,24:PRINTT5$;:LOCATE78,24:P
RINTT6$;:LOCATE0,19:PRINTT7$;:LOCATE78,1
9:PRINTT8$;:LOCATE0,4:PRINTT7$;:LOCATE78
,4:PRINTT8$;:DEFUSR=&H44:A=USR(0)
1740 RETURN
1750 '
1760 DEFUSR=&HBA00:RESTORE1780:FORI=&HBA
00TO&HBA1F:READA$:POKEI,VAL("&h"+A$):NEX
TI
1770 RETURN
1780 DATA 06,05,C5,3A,DD,F3,F5,06,34,C5,
3E,20,CD,A2,00,C1,10,F7,21,DC,F3,7E,3C
1790 DATA 77,F1,32,DD,F3,C1,10,E3,C9
1800 '
1810 GOSUB1490:GOSUB1510:LOCATE0,5:POKE&
HF3B0,78:FILES:POKE&HF3B0,80:LOCATE2,21:
LINEINPUT"Entre com o arquivo: ";A$
1820 CADIR!=ADIR!:ADIR!=DIR!
1830 BAS!=ADIR!:FORJ=1TO11:IFMID$(A$,J,1
)<>" "THENNEXTJ
1840 J=J-1:FORI=1TOJ:IFMID$(A$,I,1)="."T
HENBAS!=ADIR!+8:I=I+1
1850 IFPEEK(BAS!)=ASC(MID$(A$,I,1))THENB
AS!=BAS!+1:NEXTI:GOTO 1080
1860 ADIR!=ADIR!+32:IFPEEK(ADIR!)<>0GOTO
1830
1870 LOCATE2,21:PRINT"Arquivo não encont
rado";SPC(20):ADIR!=CADIR!
1880 CI!=PEEK(ADIR!+27)*256+PEEK(ADIR!+2
6):PGM!=BPGN!
1890 SE!=PEEK(PGN!+1)*256+PEEK(PGN!):IFS
E!<>CI!THENPGN!=PGN!+2:GOTO1890
1900 GOSUB 1490:GOSUB1510:GOSUB1920:GOSU
B1950:GOTO 1240
1910 '
1920 LOCATE 7,6:PRINT"Arquivo: ":LOCATE
35,6:PRINT"Tamanho: ":LOCATE 58,6:PRINT"
Data: ":LOCATE 2,8:PRINT"Distribuição: "
1930 RETURN
```

O PODER DA CRIAÇÃO...

COM O PROFESSIONAL PUBLISHER VOCÊ TERÁ SOLUÇÕES PROFISSIONAIS COM TECNOLOGIA 100% NACIONAL.

Finalmente o primeiro programa especializado em Desktop Publishing para MSX está ao seu alcance. Com o PROFESSIONAL PUBLISHER você poderá criar páginas como as elaboradas nos melhores editores para micros de 16 bits. Veja algumas das funções do programa:

- importação de qualquer texto em ASCII de qualquer processador de textos do mercado.
- mistura de textos e gráficos na página sem qualquer restrição.
- múltiplas bordas para a sua página com troca a qualquer instante do trabalho.
- totalmente "user-friendly" com menus iconográficos inteligentes.
- compatível a nível de shapes, telas e alfabetos com o editor Graphos 3 ou Pro.
- uso de alfabetos em formato shape com a mesma facilidade dos alfabetos comuns.
- possibilita o uso de MEGARAM-DISK para acelerar o processo de criação da página.
- acompanha discos de apoio ao usuário e manual completo de utilização.
- totalmente programado em Pascal-Assembler pela equipe de programação da Discovery Informática.
- disponível em discos de 3 1/2 e 5 1/4 pols.
- registro legal no I.N.P.I.
- garantia e suporte permanentes.

Preço: 120 BTNs

EQUIPE DE CRIAÇÃO

PROGRAMAÇÃO: Vitor Hugo P. Costa
PLANEJAMENTO VISUAL: Alberto C. Meyer
ROTINAS DE DISCO: Leonardo Beltrão

COLABORAÇÃO

Alexandre Ramires
Zelman Sirola
Wellington Silvares
Antonio R. Varella
Mário Willmersdorf

Envie cheque nominal ou vale postal à:

DISCOVERY INFORMÁTICA LTDA.
Rua da Quitanda, 19 gr. 404
Centro — Rio de Janeiro — RJ
Caixa Postal 3043 — CEP 20001

Exija este software com a etiqueta da DISCOVERY INFORMÁTICA. Ela é sua garantia de qualidade. Não deixe que o pirata engane você.

```
1940
1950 LOCATE2,20:PRINT"SLCT  - examina arquivo";SPC(3);"ESC - abandona programa";SPC(3);"CLS   - inicializa disco"
1960 LOCATE3,21:PRINT">    - avança arquivo";SPC(31);"<    - retorna arquivo"
1970 LOCATE3,22:PRINT"^    - primeiro arquivo";SPC(29);"v    - último arquivo"
1980 RETURN
```

2a PARTE DO ARTIGO: POR DENTRO DO DRIVE - ROTINAS PARA SIMULAÇÃO DO BDOS

Autor: Júlio Renato Soares Velloso
Revista: CPU (Aguia informática)

OBS: Retirado do programa: Sistema Operacional BKP DISCO Versão: 2.0

```
        ORG #9000-7

        DEFB #FE
        DEFW TNICIO
        DEFW FTNAL
        DEFW INTCTD

PHYDIO :EQU #FFA7
TNTCTD :EQU $

ROT1   :JP STCONV
ROT2   :XOR A
        JP TRQDSC
ROT3   :JP SOCONV
ROT4   :JP MNTDIR
ROT5   :JP ERRMAP
ROT6   :JP CACFAT
ROT7   :JP ATLMAP
ROT8   :JP SAVFAT
ROT9   :JP GRVNUM

TAB1   :DEFW BUEPGM
TAB2   :DEFW BUFDIR
TAB3   :DEFW DTRET
TAB4   :DLEW BUFMAP
TAB5   :DEFW BAKFAT

CPHLDE :LD A,H
        SUB D
        RET NZ
        LD A,L
        SUB F
        RET

STCONV :CP T              ; << PEGA CLUSTER DA FAT >>
        JR Z,STC00T
        LD A,(IY+2)       ; > PARTE 2 DO SEGMENTD (VSET=2)
        LD (IY+2),0
        PUSH AF
        LD A,(TY+1)
        PUSH AE
        AND #F
        LD (IY+T),A
        POP AF
        AND #F0
        RRCA
        RRCA
        RRCA
        RRCA
        LD L,A
        POP AF
        PUSH AE
        AND #F
        RLCA
        RLCA
        RLCA
        RLCA
        LD B,A
        LD A,L
        OR B
        LD L,A
        POP AF
        AND #F0
        RFLA
        RRCA
        RRCA
        RRLA
        LD H,A
        RLT
STC001 :LD A,(TY+0)       ; > PARTE 1 DO SEGMENID (VSLT=1)
        LD (IY+0),0
        LD L,A
        LD A,(IY+1)
        PUSH AF
        AND #F0
        LD (IY+I),A
        POP AF
        AND #F
        LD H,A
        RET

TRQDSC :LD HL,BUFPGM
        LD DE,T
        LD BC,#0IFC
        LD (DRTVE),A
        AND A
        CALL PHYDIO       ; > IDLNTIEICA DISLO
        LD A,(BUFPGM)
        LP #FC
        CALL Z,FS40       ; > PEGA INFORMAÇOES P/ LACE SIMPLES 40 TRTLHAS
        CP #FD
        CALL Z,FD40       ; > PLGA INFORMAÇOES P/ FACE DUPLA 40 TRILHAS
        CP #F8
        CALL Z,FS80       ; > PEGA INFORMAÇOLS P/ FACE SIMPLLS D0 TRTLHAS
        LP #F9
        CALL Z,FD80       ; > PEGA INFORMAÇOES P/ FALL DUPLA 80 TRILHAS
        LD (FORMAT).HL    ; > REGISTRA AS INFDRMAÇOLS
        LD H,0
        LD L,L
        LD (DIVS),HL
        LD L,B
        LD (SOMA),HL
        LD (TAMSLT),DE
        PUSH IY
        POP BC
        LD A,B
        LD (TRTLHAS),A
        LD A,C
        LD (TAMFAT),A
        PUSH IX
        POP BC
        LD L,C
        LD (POSDIR),HL
        LD A,(FORMAI)
        LD C,A
        LD (FORMAI2),BC
        RLT
FS40   :LD HL,#!FL        ; > FORMATAÇAO FACE SIMPLES 40 TRILHAS
        LD BC,#07A5       ; > DIFERLNÇA ENTRE RLAL E FAT 1º SETDR LIVRE
```

```
        LD DE,#2B0      ; > BYTES POR CLUSTER
        LD IY,#2802     ; > I- NUM. DE TRILHAS / Y- TAM. DA FAT
        LD IX,#0405     ; > I- TAMANHO DO DIR. / X- INICIO DE DIRETORIO
        XOR A
        RET
FD40   :LD HL,#2FD      ; > FORMATAÇAO FACE DUPLA 40 TRILHAS
        LD BC,#04BC     ; > DIFERENÇA ENTRE REAL E FAT 1o SETOR LIVRE
        LD DE,#400      ; > BYTES POR CLUSTER
        LD IY,#2802     ; > T- NUM. DE TRILHAS / Y- TAM. DA FAT
        LD IX,#0705     ; > I- TAMANHO DO DIR. / X- INICIO DE DIRETORIO
        XOR A
        RET
FS80   :LD HL,#1F8      ; > FORMATAÇAO FACE SIMPLES 80 TRILHAS
        LD BC,#0ABC     ; > DIFERENÇA ENTRE REAL E FAT 1o SETOR LIVRE
        LD DE,#200      ; > BYTES POR CLUSTER
        LD IY,#5002     ; > I- NUM. DE TRILHAS / Y- TAM. DA FAT
        LD IX,#0705     ; > I- TAMANHO DO DIR. / X- INICIO DE DIRETORIO
        XOR A
        RET
FD80   :LD HL,#2F9      ; > FORMATAÇAO FACE DUPLA 80 TRILHAS
        LD BC,#050E     ; > DIFERENÇA ENTRE REAL E FAT 1o SETOR LIVRE
        LD DE,#400      ; > BYTES POR CLUSTER
        LD IY,#5003     ; > I- NUM. DE TRILHAS / Y- TAM. DA FAT
        LD IX,#0707     ; > I- TAMANHO DO DIR. / X- INICIO DE DIRETORIO
        XOR A
        RET

SOCONV :LD HL,BUFMAP    ; > CARREGA FAT
        LD DE,1         ; > INICIO DA FAT
        LD BC,(FORMAT)  ; > FORMATAÇAO
        LD A,(TAMFAT)   ; > TAMANHO DA FAT EM SETORES
        LD B,A
        LD A,(DRIVE)    ; > DRIVE DEFAUT
        AND A           ; > MODO DE LEITURA
        CALL PHYDIO
        LD HL,#FFF
        LD (BUFPGN),HL
        LD HL,BUFPGN+2  ; > INICIO DO BUFFER A SER MONTADO
        LD (BUFPFN),HL
        LD IY,BUFMAP+3  ; > INICIO DA FAT
        LD HL,2         ; > NSET - 1o CLUSTER DO DISCO (2)
        LD (NSET),HL
        LD HL,0         ; > SNSET PARA REGISTRO DE ARQUIVO DIVIDIDO
        LD (SNSET),HL
        LD A,1          ; > VSET - 1a PARTE DO SEGMENTO
        LD (VSET),A
        LD A,(TAMFAT)
        CALL CACFAT
        LD HL,BUFMAP    ; > FINAL DA FAT
        ADD HL,BC
SOCO00 :DEC HL          ; > ENCONTRA ULTIMO CLUSTER SETADO
        LD A,(HL)
        AND A
        JR Z,SOCO00
        INC HL
        LD (FINMAP),HL  ; > REGISTRA EM FINMAP
        LD DE,BUFDIR+3  ; > VERIFICA SE FAT ESTA VASIA
        CALL CPHLDE
        JP Z,FRRMAP
        LD H,(IY+1)     ; > PRESERVA SEGMENTO DA FAT
        LD L,(IY+0)
        PUSH HL
        LD A,(VSET)
        CALL STCONV     ; > PEGA CLUSTER NA FAT
        LD A,H          ; > VERIFICA SE O 1o, E CLUSTER APAGADO
        OR L
        POP HL
        LD (IY+1),H     ; > RECUPERA SEGMENTO DA FAT PRESERVADA
        LD (IY+0),L
        JR NZ,SOCO01
        CALL SOCO09     ; > VERIFICA CLUSTER'S APAGADOS
SOCO01 :LD HL,(NSET)    ; > HL COM CLUSTER ATUAL
        CALL SOCO03     ; > COLOCA NO BUFFER
        CALL SOCO09     ; > VERIFICA CLUSTER'S APAGADOS
SOCO02 :LD DE,#FFF      ; > VERIFICA SE FOI FINAL DE ARQUIVO
        CALL CPHLDE
        JP Z,SOCO04     ; > TRATA FINAL DE ARQUIVO
```

```
        JP SOCO05       ; > ARQUIVO DIVIDIDO
SOCO03 :LD IX,(BUFPFN)  ; > COLOCA NO BUFFER O CLUSTER EM HL
        LD (IX+0),L
        LD (IX+1),H
        INC IX
        INC IX
        LD (BUFPFN),IX
        RET

SOCO04 :LD HL,#FFF      ; << TRATA FINAL DE ARQUIVO >>
        CALL SOCO03     ; > REGISTRA NO BUFFER FINAL DE ARQUIVO
        LD HL,(SNSET)   ; > TESTA SNSET P/ ARQUIVO DIVIDIDO
        LD A,H
        OR L
        CALL NZ,SOCO07  ; > TRATA FINAL DE ARQUIVO DIVIDIDO
        CALL SOCO08     ; > INCREMENTA PONTEIROS
        CALL SOCO09     ; > AJUSTA PARA PROXIMO CLUSTER SETADO DA FAT
        PUSH HL         ; > REGISTRA CLUSTER
        LD HL,(NSET)
        CALL SOCO03
        POP HL
        LD DE,#FFF      ; > VERIFICA FINAL DE ARQUIVO
        CALL CPHLDE
        JP Z,SOCO04     ; > CASO SIM, TRATA FINAL DE ARQUIVO

SOCO05 :LD DE,(NSET)    ; << VERIFICA E TRATA ARQUIVO DIVIDIDO >>
        AND A
        SBC HL,DE
        PUSH HL
        DEC HL
        LD A,L
        OR H
        POP HL
        JP NZ,SOCO06    ; > TRATA ARQUIVO DIVIDIDO
        CALL SOCO08     ; << INCREMENTA PONTEIROS >>
        JP SOCO01       ; > PROXIMO CLUSTER

SOCO06 :PUSH HL         ; << TRATA ARQUIVO DIVIDIDO >>
        LD HL,(SNSET)   ; > TESTA SE JA HOUVE QUEBRA DE ARQUIVO
        LD A,L
        OR H
        JR NZ,SOCO61    ; > CASO SIM, AJUSTA PONTEIRO
        LD HL,(NSET)    ; > CASO NAO, PRESERVA PONTEIROS
        LD (SNSET),HL   ; > SNSET COM CLUSTER DIVIDIDO
        LD A,(VSET)
        LD (SVSET),A    ; > SVSET COM PARTE DO SEGMENTO
        LD (SPONT),IY   ; > SPONT COM PONTEIRO DA FAT
SOCO61 :POP BC          ; > AJUSTA PONTEIRO
SOCO62 :CALL SOCO08
        DEC BC
        LD A,C
        OR B
        JR NZ,SOCO62
        JP SOCO01

SOCO07 :LD HL,(SNSET)   ; << RESETA SNSET P/ ARQUIVO DIVIDIDO >>
        LD (NSET),HL    ; <<     E RECUPERA OS PONTEIROS     >>
        LD HL,0
        LD (SNSET),HL
        LD A,(SVSET)
        LD (VSET),A
        LD IY,(SPONT)
        RET

SOCO08 :PUSH HL         ; << AJUSTA PONTEIROS >>
        LD HL,(NSET)    ; > INCREMENTA (NSET)
        INC HL
        LD (NSET),HL
        POP HL
        LD A,(VSET)     ; > TESTA SE VSET E 2a PARTE DO SEGMENTO
        CP 2
        JR Z,SOCO81     ; > CASO SIM, AJUSTA IY, VSET E FIM MAPA
        LD A,2          ; > CASO NAO, ALTERA VSET
        LD (VSET),A
        RET
SOCO81 :INC IY          ; > AVANÇA P/ PROXIMO SEGMENTO
        INC IY
        INC IY
```

```
        LD A,L          ;  FAZ VSET IGUAL A L
        LD (VSET),A
        LD HL,(SNSET)   ; > VERIFICA SE ARQUIVO FOI DIVIDIDO
        LD A,H
        OR L
        RET NZ          ; > CASO SIM, VOLTA
        PUSH IY         ; > CASO NAO, VERIFICA FINAL DE MAPA
        POP HL
        LD DE,(FIMMAP)
        CALL CPHLDE
        RET C           ; > CASO NAO, RETORNA
        LD HL,#FFFF     ; > CASO SIM, COLOCA BYTE DE TERMINO E ...
        CALL SOC003
        POP HL          ; ... TERMINA A ROTINA
        RET

SOC009 :LD A,(VSET)     ; << VERIFICA E TRATA CLUSTER APAGADO >>
        CALL STCONV
        LD A,H
        OR L
        PUSH AF
        CALL Z,SOC00B   ; > INCREMENTA PONTEIROS
        POP AF
        JR Z,SOC009
        XOR A
        RET

CRAMAP :LD HL,(SOMA)    ; << REGISTRA INFORMAÇOES P/ DISCO VASIO >>
        INC HL
        INC HL
        LD DE,4
        AND A
        SBC HL,DE       ; > PEGA 1o CLUSTER LIVRE
        LD (BUFPGM+2),HL; > REGISTRA EM BUFPGM
        LD (BUFDIR),HL  ; > REGISTRA EM BUFDIR
        LD HL,#200
        LD (BUFDIR+2),HL; > REGISTRA TAMANHO
        LD HL,#FFFF
        LD (BUFPGM+4),HL; > REGISTRA SINAL DE FIM DE MAPA
        LD (BUFDIR+4),HL
        LD A,#FF
        RET

MNTDIR :LD HL,DIRET     ; << CARREGA DIRETORIO DO DISCO >>
        LD DE,(POSDIR)
        LD BC,(FORMAT2)
        LD A,(DRIVE)
        AND A
        CALL FNYDIO
        LD IX,BUFDIR    ; << MONTA TABELA 'ENTRADA/TAMANHO' >>
        LD IY,BUFPGM+2
        LD H,(IY+1)     ; > 1a ENTRADA
        LD L,(IY+0)
MNTDIR1:LD (IX+1),H     ; > REGISTRA ENTRADA
        LD (IX+0),L
        LD HL,0         ; > ZERA TAMANHO
MNTDIR2:INC IY          ; > PROXIMO CLUSTER
        INC IY
        LD DE,(TAMSET)  ; > AUMENTA TAMANHO
        ADD HL,DE
        PUSH HL
        LD A,(IY+1)
        LD L,(IY+0)
        AND #F
        LD H,A
        LD DE,#FFF      ; > VERIFICA FINAL DE ARQUIVO
        CALL CPHLDE
        POP HL
        JR NZ,MNTDIR2   ; > REPETE ATE CHEGAR NO FINAL
        INC IY          ; > PROXIMO ARQUIVO
        INC IY
        LD (IX+3),H     ; > REGISTRA TAMANHO
        LD (IX+2),L
        INC IX          ; > PROXIMO REGISTRO DO PROXIMO ARQUIVO
        INC IX
        INC IX
        INC IX
        LD H,(IY+1)     ; > PEGA PROXIMA ENTRADA
```

```
        LD L,(IY+0)
        LD DE,#FFFF     ; > VERIFICA FIM DE DISCO
        CALL CPHLDE
        JR NZ,MNTDIR1
        LD (IX+1),H     ; > REGISTRA FIM DE BUFFER
        LD (IX+0),L
        RET

CACFAT :LD HL,0         ; << CALCULA EM BYTES O TAMANHO DA FAT >>
        LD DE,#200      ; > 1 SETOR POSSUI 200H BYTES
        LD B,A
CACF01 :ADD HL,DE
        DJNZ CACF01     ; > TAMANHO DA FAT = TAMFAT * 200
        PUSH HL
        POP BC
        RET

ATLMAP :NOP             ; << MONTA FAT BASEADO EM BUFPGM >>
        LD A,(TAMFAT)
        CALL CACFAT
        LD IY,BUFPGM+2  ; > IY COM INICIO DE BUFPGM
        LD IX,BUFMAP    ; > IX COM INICIO DA FAT
        PUSH IX
        PUSH IX
        POP HL          ; > HL COM 1a POSIÇAO
        POP DE
        INC DE          ; > DE COM 2a POSIÇAO
        LD (HL),0
        LDIR            ; > ZERA TODA A FAT
        LD A,(FORMAT)   ; > 1o POSIÇAO DA FAT COM BYTE DE FORMATAÇAO
        LD (IX+0),A
        LD (IX+1),#FF   ; > 2a E 3a POSIÇOES DA FAT COM FF
        LD (IX+2),#FF
        LD H,(IY+1)     ; > 1a CLUSTER DE BUFPGM
        LD L,(IY+0)
ATLM01 :LD B,H          ; > BC COM CLUSTER
        LD C,L
        LD IX,BUFMAP+3  ; > IX COM INICIO DA FAT
        DEC BC
        DEC BC
        LD A,B
        OR C
        LD A,0
        JR Z,ATLM04     ; > SE BC=2 REGISTRA
ATLM02 :AND A           ; << LOCALIZA SEGMENTO (IX) E PARTE DELE (A) >>
        JR Z,ATLM03
        INC IX          ; > PROXIMA SEGMENTO DA FAT
        INC IX
        INC IX
        LD A,-1         ; > A COM 1a PARTE DO SEGMENTO (A=0)
ATLM03 :INC A
        DEC BC          ; > DIMINUI BC
        LD D,A
        LD A,B
        OR C
        LD A,D
        JR NZ,ATLM02    ; > REPETE BC VEZES
ATLM04 :INC IY          ; > PROXIMO CLUSTER
        INC IY
        LD H,(IY+1)     ; > PEGA CLUSTER
        LD L,(IY+0)
        CALL GRVNUM     ; > GRAVA CLUSTER NA FAT
        LD DE,#FFF      ; > VERIFICA FINAL DE ARQUIVO
        CALL CPHLDE
        JR NZ,ATLM01
        INC IY          ; > AVANÇA P/ PROXIMO ARQUIVO
        INC IY
        LD H,(IY+1)
        LD L,(IY+0)
        LD DE,#FFFF     ; > VERIFICA FINAL DE BUFFER
        CALL CPHLDE
        JR NZ,ATLM01
        RET

SAVFAT :NOP             ; << SALVA FAT NOVA FASENDO BAKUP DA ANTIGA >>
        CP #FF          ; > VERIFICA SE O USUARIO DESEJA FAZER BAKUP
        JR Z,SAVF02
```

```
        LD HL,BAKFAT
        PUSH HL
        LD DE,1
        LD BC,(FORMAT)
        LD A,(TAMFAT)
        LD B,A
        LD A,(DRIVE)
        AND A
        CALL PHYDIO     ; > CARREGA FAT
SAVF01 :POP HL
        LD A,(TAMFAT)
        INC A
        LD E,A
        LD D,0
        LD BC,(FORMAT)
        LD A,(TAMFAT)
        LD B,A
        LD A,(DRIVE)
        SCF
        CALL PHYDIO     ; > SALVA FAT EM COPIA
        LD HL,BUFMAP
        LD DE,1
        LD BC,(FORMAT)
        LD A,(TAMFAT)
        LD B,A
        LD A,(DRIVE)
        SCF
        CALL PHYDIO     ; > SALVA FAT NOVA
        RET
SAVF02 :LD HL,BUFMAP
        PUSH HL
        JR SAVF01

GRVNUM :CP 1            ; << GRAVA UM CLUSTER NA FAT >>
        JR Z,GRVN01
        LD (IX+0),L     ; > 2o PARTE DO CLUSTER
        LD A,H
        AND #0F
        LD C,A
        LD A,(IX+1)
        OR C
        LD (IX+1),A
        RET
GRVN01 :LD A,L          ; > 1o PARTE DO CLUSTER
        AND #F0
        RRCA
        RRCA
        RRCA
        RRCA
        LD C,A
        LD A,H
        AND #0F
        RLCA
        RLCA
        RLCA
        RLCA
        OR C
        LD (IX+2),A
        LD A,L
        AND #0F
        RLCA
        RLCA
        RLCA
        RLCA
        LD C,A
        LD A,(IX+1)
        OR C
        LD (IX+1),A
        RET

FORMAT :DEFW 0          ; > DADO DE BC PARA CARREGAR UM CLUSTER (PHYDIO)
FORMAT2:DEFW 0          ; > DADO DE BC PARA CARREGAR O DIRETORIO (PHYDIO)
DIVS   :DEFW 0          ; > SETOR INICIAL DO DISCO
SOMA   :DEFW 0          ; > VARIAVEL QUE SOMA COM CLUSTER DA FAT PARA EN-
;                           CONTRAR CLUSTER GERAL
DRIVE  :DEFB 0          ; > VARIAVEL DO DRIVE ATUAL
TAMSET :DEFW 0          ; > TAMANHO DE UM CLUSTER
TAMFAT :DEFB 0          ; > TAMANHO DE UMA FAT EM SETORES
POSDIR :DEFW 0          ; > POSIÇAO INICIAL DO DIRETORIO
TRILHAS:DEFW 0          ; > NUMERO DE TRILHAS DO DISCO

NSET   :DEFW 0          ; > VARIAVEL QUE CONTEM CLUSTER ATUAL
VSET   :DEFB 0          ; > VARIAVEL QUE CONTEM PARTE DO SEGMENTO
SNSET  :DEFW 0          ; > SALVAMENTO DE NSET P/ ARQ. DIVIDIDO
SVSET  :DEFB 0          ; > SALVAMENTO DE VSET P/ ARQ. DIVODIDO
SPONT  :DEFW 0          ; > SALVAMENTO DO PONTEIRO DA FAT P/ ARQ. DIVIDIDO
FIMMAP :DEFW 0          ; > FIM DA FAT (BUFMAP)
BUFPFN :DEFW 0          ; > FIM DO (BUFPGM)

FINAL  :EQU $
BUFMAP :EQU $           ; > P/ COLOCAR A FAT
BAKFAT :EQU $+#600      ; > P/ COLOCAR UM BAKUP DA FAT
BUFDIR :EQU $+#C00      ; > P/ COLOCAR A TABELA (ENTRADA/TAMANHO)
BUFPGM :EQU $+#E00      ; > P/ COLOCAR A TABELA (FAT TRADUZIDA)
DIRET  :EQU $+#1700     ; > P/ COLOCAR O DIRETORIO
```

MEGARAM OU MEGA HÃ?

A expansão de memória Megaram tem se tornado, a cada dia, mais popular e muitos usuários questionam o seu funcionamento. Análises anteriores deste produto já foram feitas, mas nenhuma de caráter sério ou técnico. Apesar deste artigo se dirigir a leitores com ou sem conhecimento técnico, o último item (funcionamento interno) contém informações técnicas de como aproveitar todo o potencial da Megaram.

Grilo Falante

ORIGENS

Não se pode dizer que a Megaram teve um bom começo. A idéia foi de um técnico de São Paulo, cujo objetivo era piratear os jogos Megarom (nome devido à capacidade dos cartuchos: 1 Megabit = 128 kb), surgidos no Japão, a partir de 1985. Os jogos são carregados na Megaram de disco mas, quando o jogo é executado, ela se comporta de modo idêntico ao cartucho.

Isso permitiu aos usuários de grande poder aquisitivo (grande mesmo: a Megaram era vendida por até US$ 250) se divertirem com jogos como Nemesis 2, Super Laydock, entre outros. Mas um ponto fundamental foi esquecido: 256 kb de memória a um dólar kbyte são demais para serem usados só para joguinhos.

64 kbytes é pouca memória nos tempos de hoje. A Megaram poderia ser a saída, mas ela sofre um problema congênito: seu sistema de funcionamento é bom para jogos mas ruim para ser usado por programas aplicativos.

Até mesmo os jogos se voltaram contra a Megaram: os fabricantes japoneses utilizam, agora, um sistema de seleção para o padrão da megaram para poderem funcionar, o que nem sempre acontece.

EXISTE COISA BEM MELHOR

Os europeus e japoneses não demoraram muito para perceber que um sistema de expansão de memória era necessário, e que a expansão de slots não era a melhor saída. Foi desenvolvido um mecanismo que, como a Megaram, trabalha por chaveamento de bancos, que é muito mais flexível que o chaveamento de slots.

Esse sistema chama-se "Memory Mapper", e já vem embutido nos micros Phillips de 128 e 256 kb. No Japão, ele vem nos micros MSX 2+, bastando aos compradores trocar os chips de 64 kb por chips de 256 kb. Outra opção é o MSX-DOS 2, que precisa dessa expansão, sendo vendido em duas versões: uma sem memória e outra com 256 kb.

O MSX-DOS 2 é uma verdadeira revolução em sistemas operacionais. Contém todos os recursos do MS-DOS usado em PCs, como subdiretórios de Winchester de até 40 Mb. Além dele, vários ótimos programas europeus só funcionam com 128 ou 256 kb de memória. São programas de animação e edição gráfica, sistema de "Desktop Publishing" e, claro, vários jogos Megarom, como Aleste e Xevious.

RAMDISK

Apesar dos problemas, a Megaram é a única expansão de memória disponível atualmente, e não devemos dar-lhe a sina que seu inventor desejou: os joguinhos. O primeiro passo na profissionalização da Megaram é seu uso como disco virtual.

Um disco virtual é a simulação em memória de um disquete. Ele é visto pelo computador como mais um drive para armazenamento de arquivos, mas este disco em RAM (ou Ramdisk) trabalha muito mais rápido que um drive normal, pois a RAM é totalmente eletrônica, não contendo nenhum dispositivo mecânico ou magnético.

O Ramdisk é especialmente útil com programas que trabalham com "Overlay" (parte do programa é carregada do disco quando necessária, economizando memória) como o SuperCalc2 e o dBase 2 Plus, que podem ter seus arquivos "Overlay" no Ramdisk. Trabalhar com dados e quaisquer outros arquivos no Ramdisk também agiliza muito qualquer serviço, mas se acabar a luz ou se esquecer de gravar o resultado final em disco magnético, pode jogar fora horas de trabalho.

Em recentes conversas com a equipe da Paulisoft, foi-me mostrado um Ramdisk (carregando de disco, pois só o fabricante da expansão tem condições de colocá-lo em ROM). No entanto, ainda não encontraram um fabricante disposto a pagar preço razoável.

A pirataria gerou, no Brasil, o hábito de não dar valor ao software, e esse Ramdisk vai, por enquanto, continuar "na gaveta". A Digital Design, mais conhecida como DDX, também desenvolveu um Ramdisk que vem incluso (como ROM) na Megaram por ela fabricada, a DDX256 (MEGA-RAM-DISK).

O MERCADO ATUAL

Hoje já há vários fabricantes de Megaram no mercado, o que aumentou a concorrência e fez cair seu preço para algo como US$ 120, ou até menos. A DDX domina grande parte do mercado, pois além de ter um produto tecnicamente mais completo, conseguiu

colocá-lo à venda em grandes magazines, tendo um sistema industrial de produção.

Além da DDX, há oito outros fabricantes semi-industriais e caseiros. O principal problema é que todos esses fabricantes trabalham com o mesmo projeto, exatamente igual à primeira Megaram produzida, que apresenta graves falhas de projeto digital, além de usarem chips de memória de baixa qualidade comprados em saldos. A falta de respeito pelo usuário chega ao ponto de sequer fornecer um manual de uso e instalação. O ditado "barato sai caro" nunca esteve tão atual. A guerra entre esses fabricantes tem trazido progressivas reduções de preço, qualidade e vendas.

Eu torço para que outras empresas entrem no mercado de Megarams, antes que ele se torne monopolizado (em janeiro, a DDX elevou o seu preço em US$ toda semana). Indústrias como Gradiente e DMX estão perdendo dinheiro, deixando de entrar neste mercado de potencial crescente.

Eu espero também que as empresas que já estão no mercado parem de usar táticas comerciais desonestas, como um certo fabricante de kits 2.0 (por sinal, ruins) que diz que Megaram de outro fabricante explode (!) o micro por eles transformado, e que só a Megaram deles funciona corretamente. Isto é totalmente FALSO, a não ser que a partir de agora esse fabricante inclua um "detector de Megaram dos outros" em seu kits 2.0, para que só a deles funcione.

FUNCIONAMENTO INTERNO

A Megaram tem seus 256 kb divididos internamente em 32 bancos de 8 kb. A idéia do sistema de bancos é que cada banco pode ser selecionadol em várias posições de memória (inclusive ao mesmo tempo), ao contrário dos slots, em que cada página tem endereço fixo.

O que determina a posição de uma informação na Megaram é o número do banco (0 a 31) e o deslocamento (0 a 1FFFh) em relação ao início do banco, não do endereço do Z80,que se usa para acessar a memória. Os endereços são apenas a janela de comunicação com a CPU. A Megaram tem quatro seletores de bancos, um para cada segmento de memória do Z80:

segmento 1: 4000h-5FFFh
segmento 2: 6000h-7FFFh
segmento 3: 8000h-9FFFh
segmento 4: A000h-BFFFh

Em cada segmento pode estar selecionado qualquer banco, e um banco pode estar ativo em mais de um segmento, mas, nesse caso, qualquer alteração no segmento se refletirá nos outros.

O sistema de seleção de bancos é idêntico aos dos cartuchos Megaram antigos. Como nos cartuchos não há RAM, qualquer acesso de escrita em um segmento é entendido como seleção de banco, não importando o endereço. Na leitura, o funcionamento é normal.

Na Megaram foi necessário introduzir um método para alteração do conteúdo dos bancos. Há uma porta (8Eh) que chaveia entre dois modos de funcionamento: Megarom e RAM. A leitura dessa porta chaveia para RAM (o valor lido não tem significado) e escrevendo-se nessa porta qualquer valor, a Megaram fica em modo Megarom.

A seleção de bancos só pode ser feita no modo Megaram, e a gravação de dados na Megaram no modo RAM. Antes de fazer qualquer destas operações, tenha certeza qual é o modo ativo naquele momento.

A Megaram da DDX, com Ramdisk, contém, também, uma porta para o Ramdisk. Qualquer acesso (leitura ou escrita) à porta 8Fh seleciona a ROM do Ramdisk, e qualquer acesso à porta 8Eh desabilita a ROM e seleciona a Megaram no modo apropriado.

Quando ligada ou após um reset por hardware (através de botão) a Megaram fica em modo Megarom ou modo Ramdisk, se for da DDX. Por isso, quando a Megaram estiver num slot anterior ao slot da RAM, se a Megaram estive em modo RAM e for dado um reset por software (programa ou comando), ela será identificada como RAM, o que causará o não funcionamento do disk-drive e conseqüências imprevisíveis para o resto do sistema, se for selecionado o modo Megaram.

O leitor deve ter percebido que não é muito simples usar a Megaram, principalmente em BASIC. É tácit trabalhar com a Megaram em Assembly, pois podemos definir onde localizar o programa, evitando problemas com o chaveamento de slots necessários para acessar a Megaram. Se o chaveamento fosse só para leitura, tudo seria mais simples. Esse é o caso do "Memory Mapper", onde os bancos são chaveados por acesso às portas FCh-FFh.

A saída é usar pequenas sub-rotinas em Assembly, que façam, por exemplo, cópia de uma área da Ram para um banco da Megaram e vice-versa ou, caso disponível, usar o Ramdisk, lendo e gravando arquivos com os comandos que toda linguagem oferece. Isto é, sem dúvida, o método mais prático de se trabalhar com a Megaram, além de permitir a um usuário que não a possua usar um disco como expansão de memória, apesar de ser muito mais lento.

Meu nome é Grilo falante pois eu quero ser a consciência que os fabricantes não tem.

SPRITES E ROTINAS DE VÍDEO

Márcio Machado de Moura

No número passado (CPU nº 15), analisamos a VRAM quanto a sua estrutura e modos de operação. Neste número, veremos o mecanismo de funcionamento dos SPRITES, além de começarmos a apresentar rotinas de alta performance para leitura e gravação da VRAM.

Antes de começarmos, porém, gostaria de me desculpar por alguns erros encontrados no artigo passado (LEITURA E GRAVAÇÃO DA VRAM), que mesmo não atrapalhando a compreensão dos conceitos apresentados, podem ter assustado alguns leitores.

Não vou me deter em explicações da origem dos erros, já que várias vezes este assunto esteve em pauta nos mais diversos tipos de literatura. Só posso dizer que a lei de MURPHY, que em revista se traduz como: "Não adianta revisar um texto 500 vezes, pois os erros só serão vistos após a impressão do mesmo", é implacável. Mesmo assim procuraremos sempre apresentar artigos livres de erros.

ERRATA

- *A citação de MSX 1.0, diz respeito à qualquer MSX versão 1, que aqui no Brasil, recebe subdivisões como: 1.0, 1.1, 1.5, etc.*
- *Na página 18, segunda coluna, quarta linha, temos uma divisão meio absurda. A conta certa é: (256 / 32 = 8).*
- *Na página 19, primeira coluna, terceira linha, foi dito que podíamos contar com 512 pontos na horizontal no modo SCREEN 2. O certo, é 256 pontos.*

SPRITES

O primeiro passo para compreensão dos SPRITES, é a desmistificação desta técnica, que as vezes, ao receber explicações ilustrativas, acaba confundindo mais do que elucidando.

Imaginemos portanto, os SPRITES como um banco de figuras, muito semelhante ao banco de caracteres. Sua primeira diferença está no tamanho, já que podem ser apresentados nos seguintes formatos:

- **8 x 8 pontos (8 bytes) – 1 pixel p/ponto:**
- **16 x 16 pontos (32 bytes) – 1 pixel p/ponto:**
- **8 x 8 pontos (8 bytes) – 4 pixel p/ponto:**
- **16 x 16 pontos (32 bytes) – 4 pixel p/ponto:**

Não é muito difícil a comparação dos sprites com caracteres, pois a sua construção na memória de vídeo, é idêntica ao banco de imagens de caracteres, ocupando também 2 Kbytes. No caso dos SPRITES maiores, (32 bytes) temos na verdade a ocupação de 4 caracteres, diminuindo assim o número de imagens de SPRITES possíveis, quando usado neste modo. O grande segredo, que diferencia os SPRITES de caracteres comuns, está em dois fatores: Transparência e posição fixa.

ÁREA DE FORMAÇÃO DE SPRITES

Como dito, esta área ocupa 2 Kbytes da VRAM, localizada a partir do endereço 3800H, em todos os modos de SCREEN que possibilitam o uso de SPRITES (modos 1, 2 e 3). Por não haver nenhum padrão pré-estabelecido em ROM, a ÁREA DE FORMAÇÃO DE SPRITES é inicializada com 00H em todas as posições de memória. Para a definição de SPRITES, existem comandos específicos no interpretador BASIC, que podem ter às vezes um uso um tanto quando confuso.

O desenho criado com SPRITES, obedece o padrão de desenho usado para caracteres, ou seja, os uns de cada byte serão representados no vídeo com a cor estabelecida como cor de frente, salientando apenas, que os bytes zero não aparecerão, dando assim a noção de transparência da figura. No caso dos SPRITES de 16 x 16 pontos, a técnica do desenho é a mesma, usando porém 4 blocos (caracteres), para formação da imagem, obedecendo a seguinte ordem:

- **1º Bloco: Canto superior esquerdo;**
- **2º Bloco: Canto inferior esquerdo;**
- **3º Bloco: Canto superior direito;**
- **4º Bloco: Canto inferior direito.**

Esquematizando a distribuição acima, teremos a seguinte geração de imagem do SPRITE de 16 x 16 pontos:

1º	3º
2º	4º

A seleção do modo de 8 x 8, nos pos-

sibilita a criação de 256 imagens, mesma quantidade do banco de caracteres (2048 / 8 = 256). Na hipótese de ser escolhido o modo de 16 x 16, teremos a redução para 64 imagens, já que serão usados 4 blocos de 8 bytes para cada imagem (2048 / 32 = 64).

ÁREA DE LOCALIZAÇÃO DE SPRITES

Nesse momento, a nossa comparação com o tratamento de caracteres, deve ser deixada para trás. A ÁREA DE LOCALIZAÇÃO DE SPRITES, ocupa 128 bytes de VRAM, do endereço 1B00H até 1B7FH, sendo que são utilizados 4 bytes sequenciais, para cada SPRITE enviado para a tela. Isto nos dá a possibilidade de termos até 32 SPRITES no vídeo (128/4 = 32), havendo porém certas restrições na geração dos SPRITES de forma horizontal, que serão discutidos mais tarde.

Enviar um SPRITE para o vídeo, significa apenas escrevermos dados em algum bloco de 4 bytes desta área. A definição de cada bloco, obedece a seguinte estrutura:

1º Byte: Linha da aparição do SPRITE;
2º Byte: Coluna da aparição do SPRITE;
3º Byte: Nº da imagem que irá aparecer;
4º Byte: Código da cor assumida pelo SPRITE.

O quarto byte, além de conter o código da cor, serve para indicar a situação de rolamento do SPRITE. Isto é possível, já que o código da cor, ocupa apenas os 4 bits menos significativos do byte, ficando os bits 3, 4 e 5 sempre com o valor zero, e o bit mais significativo (bit 7), servindo como o Flag citado. A aparência deste 4º byte, assume a seguinte forma:

F000CCCC

Onde F representa o Flag, e CCC o código da cor, que pode assumir de 0 (0000 binário) até 15 (1111 binário).

HIERARQUIA DE APARIÇÃO

No momento em que os SPRITES aparecem na tela, passando por cima da imagem existente, fica a pergunta: E quando um SPRITE passar por cima de outro?

A resposta está na hierarquia de aparição, ou seja, se numerarmos a ÁREA DE LOCALIZAÇÃO DO SPRITE de 0 a 31, teremos uma ordem de aparições, que corresponde a própria numeração da área. Isto significa, que o código da imagem de qualquer SPRITE, enviado para o bloco de localização zero, fica por cima de qualquer outra imagem, mesmo a de outro SPRITE. Desta foram, temos como que 32 níveis de superposição de SPRITES, nunca havendo mais que um SPRITE por nível. Por isso, temos aquela impressão, em jogos, de que objetos animados passam uns por cima de outros.

Outra dúvida que pode existir, é o que acontece com um SPRITE, quando este muda de posição. É fácil de entendermos o princípio, bastando para tanto, encararmos o bloco de localização que contém um código qualquer de imagem de SPRITE, em seu 3º byte, como um padrão dedicado àquele SPRITE, enquanto não alterarmos este 3º byte. A variação dos dois primeiros bytes, dedicados às coordenadas vertical e horizontal do SPRITE, faz com que a imagem desapareça da posição corrente (sem alterar a imagem existente por baixo), e reapareça nas novas coordenadas.

LOCALIZAÇÃO NO VÍDEO

Em qualquer modo de SCREEN, permitido para uso de SPRITES, obedeceremos a escala de posições mais alta da versão 1 dos equipamentos MSX, ou seja, 192 pontos verticais por 256 pontos horizontais. Escrevermos valores nos bytes de coordenadas de um bloco de localização, significa posicionarmos o bit mais alto, e mais a esquerda da imagem selecionada, na posição especificada pelos dois primeiros bytes do bloco de localização.

Um detalhe importantíssimo, quanto a especificação de coordenadas, reside no fato de termos uma defasagem na escala vertical padrão de 0 à 191, de uma posição para cima. Sendo assim, a primeira linha útil de vídeo, para posicionamento de SPRITES, é a posição -1 (FFH), e a última fica sendo a posição 190 (BEH). Valores menores do que -1, servem para rolamento vertical da imagem do SPRITE.

O posicionamento horizontal respeita a escala padrão de 0 até 255, havendo problemas no rolamento. Para isso temos o Flag do 4º byte, que possibilita um recuo instantâneo de 256 pontos, quando alcançamos a posição mais à direita da tela. A condição que indica este comportamento ao VDP, é a colocação do valor 1, no 1º bit do 4º byte do bloco de posição que se deseja associar ao rolamento:

- **1000CCCC = "Rola" à direita;**
- **0000CCCC = Não "rola" à direita.**

Diante do exposto, e para que o tratamento de SPRITES fique mais claro, vamos dar um exemplo da utilização destas áreas, enviando uma imagem de um SPRITE, para o centro da tela, posicionando-o por cima de todas as figuras, inclusive de outros SPRITES. O meio de realizarmos isto, não será associado a nenhuma linguagem, usando apenas o conceito de ESCREVERMOS NA MEMÓRIA DE VÍDEO, o que é possível de ser feito em inúmeras linguagens. Desta forma, será possível, aos que preferem elaborar programas em PASCAL, BASIC ou C, criarem suas próprias rotinas de SPRITES, sem a necessidade do uso da linguagem ASSEMBLY. Passemos então à apresentação do exemplo.

USANDO SPRITES

O primeiro passo, para o envio de um SPRITE para o vídeo, é a definição do seu desenho ou imagem. Para tal, escolhi uma figura simples, pois o objetivo é puramente ilustrativo. A representação será em código binário, formado por 64 pontos (modo de SPRITE de 8 x 8), o que nos dá um total de 8 bytes:

1º Byte: 11111111
2º Byte: 11000011
3º Byte: 10100101
4º Byte: 10011001
5º Byte: 10011001
6º Byte: 10100101
7º Byte: 11000011
8º Byte: 11111111

O desenho, como podemos observar, é um quadrado com um X no meio. Obtida a imagem, temos que associá-la a um dos 256 códigos de SPRITE, na ÁREA DE FORMAÇÃO DE SPRITES. Para isso, temos de escrever, na ordem, cada um dos 8 bytes em um bloco da área. Para melhorarmos nosso exemplo, não usarei o primeiro bloco, colocando a nossa figu-

ra no 23º bloco (número totalmente aleatório).

Não devemos nos esquecer, que o 23º bloco, é na realidade o bloco 22, devido ao fato de termos os blocos numerados de 0 à 255. Sendo assim, o bloco 0 ocupa os bytes 0, 1,......,6, 7 a partir do endereço base da ÁREA DE FORMAÇÃO DE SPRITES. O bloco 1 vai do byte 8 ao byte 15, e assim sucessivamente. Portanto, o início do bloco 22 pode ser obtido, pela multiplicação do número do bloco, por oito:

22 x 8 = 176

Este valor deve ser somado então ao endereço base da ÁREA DE FORMAÇÃO DE SPRITES. Já que o valor deste endereço está expresso em hexadecimal, façamos primeiro, a conversão do valor obtido para esta base:

176 decimal = B0 hexadecimal

Podemos agora achar o ponto certo, na VRAM, que corresponde a posição inicial do 23º bloco na ÁREA DE FORMAÇÃO DE SPRITES:

3800H · Endereço base da área de formação
+ B0H · Deslocamento de 23 blocos
38B0H · Posição real do bloco 22

Descoberto o endereço, colocamos a imagem do SPRITE, no bloco correspondente, lembrando que estamos trabalhando no modo de imagem de 64 pontos (8 x 8):

- **38B0H < - 11111111 (FFH);**
- **38B1H < - 11000011 (C3H);**
- **38B2H < - 10100101 (A5H);**
- **38B3H < - 10011001 (99H);**
- **38B4H < - 10011001 (99H);**
- **38B5H < - 10100101 (A5H);**
- **38B6H < · 11000011 (C3H);**
- **38B7H < - 11111111 (FFH).**

Os valores entre parênteses, colocados à direita dos valores expressos em binário, são a representação dos bytes na base hexadecimal. Após serem colocados os bytes que representam a imagem do SPRITE, em um dos blocos de formação, temos que enviá-los para o vídeo. Isto é feito através da ÁREA DE LOCALIZAÇÃO DE SPRITES, conforme já vimos na teoria.

A ordem de escrita dos 4 bytes formadores do bloco de localização, não precisa ser necessariamente sequencial, pois às vezes precisamos mudar apenas a posição vertical do SPRITE, ou mesmo a cor. Vamos então definir quais os endereços que correspondem aos 4 bytes do bloco escolhido. Já que escolhemos a posição mais à frente da tela (sobre todas as figuras), os 4 bytes são exatamente os primeiros à partir do início da área:

- **Endereço 1B00H: Posição vertical;**
- **Endereço 1B01H: Posição horizontal;**
- **Endereço 1B02H: Número da imagem;**
- **Endereço 1B03H: Cor do SPRITE.**

Passemos então ao preenchimento dos bytes que formam o bloco de localização escolhido, à partir da posição de vídeo.

Considerando que nossa figura tem 8 pontos de largura por 8 pontos de altura, e que a tela tem 256 pontos horizontais por 192 pontos verticais, a posição, que deixaria nosso SPRITE, exatamente no centro corresponde à:

- **(256 - 8) / 2 = 124 (posição horizontal);**
- **(192 - 8) / 2 = 92 (posição vertical).**

Isto significa que teremos 124 colunas, antes do 1º byte do SPRITE, e 92 linhas acima da mesma figura. Sendo assim as coordenadas, na escala de SPRITES, seriam:

- **Coluna: 124 (Escala normal);**
- **Linha: 91 (Escala normal - 1).**

Na coordenada vertical, foi subtraída uma linha, devido ao fato do início da escala de SPRITE começar por -1, como já explicado anteriormente. Obtidos os valores das coordenadas de localização do SPRITE, basta escrevê-los nos bytes indicados:

- **1B00H ← 91;**
- **1B001H ← 124.**

Para que o SPRITE apareça, temos agora que associarmos ao 3º byte do bloco de localização, o número do SPRITE selecionado, sem o que teremos a imagem do 1º SPRITE, já que o valor inicial de toda a ÁREA DE LOCALIZAÇÃO DE SPRITES é zero. Novamente só temos de colocar no endereço de memória selecionado, o número da nossa imagem:

- **1B02H ← 22.**

Por enquanto, nosso SPRITE, apareceria com a cor da borda, já que ainda não informamos qual a cor escolhida. Supondo que quisessemos, por exemplo, que nossa imagem aparecesse na cor vermelha, bastaria escrever tal cor no byte indicado:

- **1B03H ← 6.**

Reparem que o bit 7, correspondente ao Flag de rolamento, terá valor igual a zero, pois não nos interessa, nesta situação, realizarmos rolamento horizontal. Após termos preenchido os 12 endereços de memória indicados, com os valores estabelecidos, obtivemos a imagem do SPRITE na tela. Se por ventura, agora, quisessemos deslocar para cima nosso SPRITE em 10 linhas, bastaria escrevermos no endereço de memória 1B00H o valor 81 (91 - 10), o que causaria o desaparecimento da figura, das coordenadas: L = 91; C = 124 e o reaparecimento nas coordenadas: L = 81; C = 124.

Acredito que conseguimos abranger toda a estrutura útil da VRAM. Finalmente podemos passar para a parte que muitos devem estar aguardando: Como ler e gravar na VRAM, através das portas de I/0, associadas ao VDP.

PORTAS DE LEITURA E GRAVAÇÃO

Existem apenas duas portas, associadas ao VDP, nos equipamentos MSX versão 1: porta 98H e porta 99H. Através delas, todas as operações de leitura e gravação, assim como a programação dos modos da VDP, podem ser realizadas. No presente artigo, não daremos enfoque aos comandos de programação do VDP, deixando isso para nosso próximo artigo. Ficaremos apenas com os algorítimos e rotinas, que possibilitam ler e gravar dados na VRAM, o que nos já garantirá uma performance muito superior aquela, quando utilizamos as rotinas padronizadas do BIOS (Sistema Básico de Entrada e Saída).

Dividimos as operações de leitura e gravação, em duas partes:

a) Posicionamento do endereço inicial do bloco;

b) Leitura ou gravação do bloco da VRAM.

Na realidade, o tratamento de I/0 (entrada/saída) da VRAM, é muito simples.

Só o que temos que fazer é, primeiro, indicarmos ao VDP qual o endereço inicial da nossa operação de I/O, seguido do tipo de manipulação, ou seja, se de leitura ou gravação. Após isso, basta lermos, via instrução ASSEMBLY IN o conteúdo do endereço, e seus subsequentes valores, ou via instrução OUT, gravarmos os valores à partir deste endereço.

Reparem que não há necessidade de enviarmos ao VDP o valor de cada endereço que será lido ou gravado, desde que os mesmos estejam organizados de forma sequencial, isto é, após prepararmos uma instrução de leitura ou gravação, posicionando a VRAM em determinado endereço, qualquer operaçãp de I/O incrementa automaticamente o endereço a ser lido ou gravado, fazendo assim com que não haja necessidade de um novo reposicionamento, para próxima operação, quando a mesma for realizada no endereço subsequente.

A forma de "startarmos" o VDP em derminado endereço da VRAM, ou seja, posicionamos a VRAM em determinado ponto, é realizada através do envio de 2 bytes pela porta 99H. Mesmo para condição de leitura, a operação de START da VRAM, é feita por duas instruções OUT, associadas à porta 99H. O que indica a operação que será realizada (Leitura ou Gravação), é o estado do bit 6, do segundo byte enviado pela porta. Considerando que sempre trabalhamos com endereços "ao contrário", devemos passar primeiro, pela porta 99H, a parte mais baixa do endereço, seguido então, da parte mais alta. A condição de LEITURA da VRAM, é feita com a colocação de 0 no bit 6 da parte mais alta do endereço (2º byte passado), e a condição de ESCRITA, é feita com a colocação de 1 no mesmo bit 6.

Não há necessidade de nos preocuparmos com esta manipulação do bit 6, pois o maior endereço, possível de ser escrito na VRAM dos MSX 1, é 37FFH (16383 decimal), que em binário ficaria assim:

00110111 11111111

Como podemos ver, o bit 6 da parte mais alta do endereço, não é atingido pelo último valor de memória da VRAM. Passemos então para a rotina que posiciona a VRAM, a partir de determinado endereço. É óbvio que deverá fazer parte dos valores passados para rotina, o tipo de operação que será realizada a partir do endereço selecionado: se de leitura ou de gravação.

LOCATE

Valores de Entrada:

Registrador A: Tipo da operação:

0 = LEITURA;
1 = GRAVAÇÃO.

Registrador HL: Endereço da VRAM:

Mínimo = 0000H;
Máximo = 37FFH.

```
LOCATE: PUSH BC
        LD C,99H    ; Valor da porta
                    ; da VDP
        OUT (C),L   ; Escreve byte mais
                    ; baixo do endereço
        CP 00H      ; Testa tipo de
                    ; operação
        JR Z,LE     ; Pula SET do bit 6
                    ; se for leitura
        SET 6,H     ; SETA bit 6 para
                    ; indicar gravação
LE:     OUT (C),H   ; Escreve byte mais
                    ; alto do endereco
        POP BC
        RET         ; Retorna
```

Como podemos ver, esta rotina serve também para selecionar endereços a serem lidos da VRAM, embora só se utilize de instruções OUT. No caso das rotinas de leitura e gravação, vamos mostrar inicialmente uma rotina que migra blocos de dados, entre a memória principal e a VRAM. Novamente vamos ter um byte que indica qual o tipo de operação realizada, ou seja, se vamos gravar dados na VRAM (RAM → VRAM), ou se vamos ler dados da VRAM (VRAM → RAM).

RAMVRM

Valores de entrada:

Registrador A: Tipo da operação:

0 = VRAM → RAM;
1 = RAM → VRAM.

Registrador HL: Fonte dos dados.
Registrador DE: Destino dos dados;
Registrador BC: Tamanho do bloco.

```
RAMVRM: CP 00H          ; Testa se
                        ; Leitura da
                        ; VRAM
        JR Z,LEVRM      ; Desvia para
                        ; VRAM <- RAM
        EX DE,HL        ; Troca fonte
                        ; com destino
        CALL LOCATE     ; posiciona
                        ; endereco da
                        ; VRAM

LERAM:  LD A,(DE)       ; Lê dado da
                        ; RAM
        OUT (98H),A     ; Escreve dado
                        ; na VRAM
        INC DE          ; Incrementa
                        ; bloco
        DEC BC          ; Decrementa
                        ; contador
        LD A,B          ; Testa se
                        ; contador = 0
        CP C
        JR NZ,LERAM     ; Continua se
                        ; contador <> 0
        RET             ; Retorna

LEVRM:  CALL LOCATE     ; Posiciona
                        ; endereco da
                        ; VRAN

LOOP2:  IN A,(98H)      ; Lê dado da
                        ; VRAM
        LD (DE),A       ; Escrece dado
                        ; na RAM
        INC DE          ; Incrementa
                        ; bloco
        DEC BC          ; Decrementa
                        ; contador
        LD A,B          ; Testa se
                        ; contador = 0
        CP C
        JR NZ,LOOP2     ; Continua se
                        ; contador <> 0
        RET             ; Retorna
```

A permuta dos valores de HL e DE, se faz necessária, devido ao fato de termos como pré-requisito da rotina LOCATE, o endereço da VRAM no registrador HL. Também devo alertar, que a rotina RAMVRM só funciona em conjunto com a rotina LOCATE.

CONCLUSÃO

As rotinas apresentadas, podem ser alteradas com grande facilidade pelos leitores que já dominam a linguagem ASSEMBLY com uma certa intimidade. No próximo artigo, apresentaremos rotinas dedicadas à outras maneiras de administrarmos a VRAM, atendendo assim, também aqueles que ainda encontram certas dificuldades na programação ASSEMBLY.

Também começaremos a análise dos comandos do VDP, que possibilitam a especificação dos modos de SCREEN, tipos de SPRITE, definição de cores, etc.

Desta forma pretendo, com a continuidade dos artigos dedicados à programação em linguagem de máquina, proporcionar aos leitores, técnicas e conhecimentos que lhes ofereçam condições de produzirem programas mais bem elaborados, ingressando assim, muitos na programação de Software Básico. Área um tanto quanto carente na realidade do nosso país.

SSC - SOFT SIMILADOR DE CARTUCHO

"Com o lançamento dos novos modelos PLUS, de micro computadores MSX da Gradiente, surge uma controvérsia entre usuários e fabricante no que diz respeito a compatibilidade de programas. O programa SSC compatibiliza alguns programas inexequível ao PADRÃO MSX."

Carlyto Zamith Oliveira

A palavra "COMPATIBILIDADE", usada frequentemente nos meios da informática, traz à tona uma controvérsia entre soft-houses e hard-houses, usuários e fabricantes. Em 1985, quando a Gradiente e a Sharp aterrizaram seus MSX's no Brasil, surgiu a primeira discussão sobre a compatibilidade dos micros MSX. No EXPERT-Gradiente, quando se escrevia "CORAÇÃO DE MÃE", o HOTBIT - Sharp reproduzia "CORA' ÔO DE MÔE". Quem era o incompatível? Quem desrespeitou o PADRÃO MUNDIAL MSX? A SHARP ou a GRADIENTE? De fato, nenhum dos dois fabricantes errou, porque o PADRÃO obriga uma padronização apenas dos caracteres alfa-numéricos e símbolos gráficos. Ao contrário do que se pensa, eles acertaram quando se uniram para criar uma nova tabela de caracteres denominada BRASCII (BRAsil Standard Code for Information Interchange). Hoje, o padrão BRASCII é uma realidade, um ponto a favor para os usuários, uma segurança de compatibilidade dos futuros MSX, que por ventura venham a ser lançados no Brasil.

Algum tempo depois, o pioneirismo vem do nordeste. Uma desconhecida e pacata empresa desponta, em meios a insolência e desapontamento, com seu irradiante Disk-Drive 5 1/4 FS com a CDX-2, uma controladora de drive, considerada, na época, a menor do mundo e uma das mais modernas. Só que ela trazia um probleminha insignificante e que mais tarde explodiria com força total, em cima dos usuários menos avisados: esta controladora, pioneira, usava as portas de I/O (DO a D7) para se comunicar com o drive. Esqueceram os projetistas daquela interface de olhar a pág. 31 do Manual de MSX HARDWARE SPECIFICATIONS release 5.0 – de 14 de novembro de 1985 – da MICROSOFT, onde dizia que os endereços de I/O DO-D7 não poderiam ser utilizados, porque o PADRÃO MSX previa o uso de múltiplas controladoras de drive. Seguindo mais adiante, relatava o manual: "Portanto, recomendamos o uso de endereços de memória", exatamente o que fez o LEOPARD e a SHARP, tempo depois, quando lançou sua controladora de drive HB6000. Temos hoje dezenas e dezenas de programas genuínos nacionais, bem elaborados, com qualidade internacional, mas que funcionam, lamentavelmente, somente nas controladoras do PADRÃO SOL.

O terceiro grito de "INCOMPATIBILIDADE" ocorreu agora, com os novos micros PLUS da Gradiente, que alocam sua RAM de 64Kbytes em slot secundário. Os milhares de games existentes no Brasil, convertidos de uma maneira frenética e inconclusa, do cartucho original importado para o disco/fita, geram tremendas dores de cabeça aos usuários na sua maioria. É que os "booteiros", programadores de BOOT (BOOT é um programinha que fica no final do jogo, responsável pela sua execução), fizeram o serviço pela metade. Sabiam eles, ou não sabiam, que o computador MSX poderia expandir até 16 slots e que em qualquer um deles, com exceção do slot 0-0, a RAM poderia se alocar. A preguiça mental ou o baixo nível de conhecimento técnico foram o suficiente para que 99,9% dos "joguinhos" não funcionassem num micro que tivesse uma concepção moderna de slots, como é o caso da série PLUS e de todos os MSX2/MSX2+ existentes no mundo. Este é o preço que nós pagamos pela política autoritária da SEI, que tenta desenvolver a tecnologia nacional por decreto. A resposta está não no "software" dos computadores mas nos cérebros dos que os usam.

Como se vê, certas "incompatibilidades" são geradas por programadores ás vezes bem intencionados mas mal informados. É necessário que as Softhouse tomem consciência e respeito ao usuário, ao desenvolver e comercializar um programa. O progamador deve depurar sua arte nas mais diversas configurações possíveis e estabelecer conceitos de adaptações. Um programa, por exemplo, escrito para trabalhar em SCREEN 2 (tela de alta-resolução do MSX1), pode facilmente adaptar-se por si, para trabalhar na SCREEN 5 (tela de média resolução do MSX2), ganhando velocidade e tratamento de cores bit-a-bit.

O MSX2 foi anunciado em maio de 1985 como uma atualização "compatível" do MSX. Programas criados no ambiente MSX podem ser executados no MSX2 sem qualquer modificação, mesmo aqueles programas em linguagem assembler. Programas e arquivos guardados em cassetes ou disco podem ser utilizados sem alteração. Características adicionais no sistema MSX2 foram improvisadas na tela, altíssima resolução, mais cores disponíveis e grande velocidade gráfica. Um circuito de relógio alimentado por bateria interna e uma RAM-DISK foram também adicionados. Uma tabela onde procurei comparar o PADRÃO MÍNIMO PARA O MSX2 e o PADRÃO MÍNIMO PARA O MSX1, poderá ser encontrada neste artigo.

Muito se fala sobre os programas que rodam no EXPERT e HOTBIT e que não rodam na SÉRIE PLUS, e poucas soluções foram divulgadas. Na verdade, quase todas elas trazem difíceis caminhos para serem seguidos pela maioria dos usuários. Instruções como puxa daqui, desloca pra lá, digita linguagem de máquina, monta, grava, sobrepõe o boot antigo, verifica se é 16K ou 32K, etc., são sempre um calvário para serem executados corretamente.

O programa SSC – Soft Simulador de Cartucho, apresentado neste artigo, tem como objetivo corrigir os "semi-bools" dos programas em disco que foram concebidos originalmente em cartucho. O programa solicita do usuário o tamanho do programa, que poderá ser 16K ou 32K (Os programas de 32K são facilmente identificados por terem dois blocos de 16k)

Aplicativos Cibertron

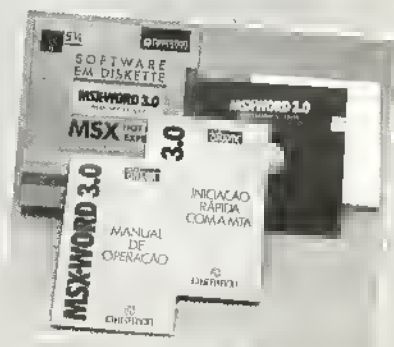
MAXIDADOS
O mais completo e sofisticado Banco de Dados em cartucho para a Linha MSX. Permite seu uso em equipamentos que tenham sua memória ampliada até 1 **MBYTE DE RAM**.
Opera com K7 ou Diskete.

MSX-WORD 3.0
Poderoso processador de textos destinado ao uso doméstico ou profissional,
permitindo armazenamento de até 480 linhas.
Inclui busca de palavras, movimentação de blocos, reformulação de parágrafos,
blocagem, definição de margens, etc.

MEGA ASSEMBLER
Permite a criação, edição e cópia de cartuchos para MSX. Podem ser criados programas em Assembler, incluindo gráficos e tabelas de caracteres.
Opção de acesso a qualquer SLOT do MSX.

OUTROS APLICATIVOS CIBERTRON
Diskete: Planilha 2.0
• Controle de Estoque
Fita K7: MSX WORD
• Assembler e Desassembler
• Planilha de MSX
• Banco de Dados.

Rua Conselheiro Saraiva, 838 · Santana
CEP 02037 · São Paulo · Capital
Telefone (011) 298-3299

REVENDAS:
Mesbla • Mappin • Filcril **São José dos Campos** - SP: Igres Infor. Tel.: (0123) 21-0321
RJ: Riosoft · Tel.: (021) 264-3726 **RS**: Sj · Infor. · Tel.: (0532) 25-9906 • Prólogos · Tel.: (0512) 22-5803

	MSX2	MSX1
CPU	Z80A ou equivalente (clock 3.579545Mhz)	
MEMORIA — ROM	48K (MSX-BASIC 2.X) MAIN-ROM 32K SUB-ROM 16K	32K (MSX-BASIC 1.X) MAIN-ROM 32K
MEMORIA — RAM	64K ou mais	8K ou mais
MEMORIA — VRAM	64K ou 128K	16K
LSI PARA O VDP	V-9938 (MSX-VÍDEO)	TMS9918 ou equiv.
DATA CORDER	FSK 1200/2400 bauds	
PSG	saída 8 oitavas tri-acorde (AY-3-8910)	
TECLADO	Alfa-numéricos e símbolos gráficos	
DISK-DRIVER (*)	Baseado no formato MS-DOS	
IMPRESSORA	Paralela 8 bits	(*)
CARTUCHOS ROM	Barramento I/O	
JOYSTICK	2	1 ou 2 (*)
RELÓGIO INTERNO	PADRÃO	(*)
RAMDISK	PADRÃO	Diferente p/cada marca

(*) Opcional

e, em seguida, o nome do(s) programa(s) ou bloco(s) e mais nada. O resto é com ele. Todo aquele emaranhado de instruções são automatizados pelo micro. O boot gerado pelo programa SSC (ver o programa-base em assembler), além de procurar a RAM nos 16 slots possíveis, transferência e execução do jogo, desliga qualquer drive que opere dentro das normas do MSX, como é o caso do DD-PLUS, SHARP HB6000, LEOPARD e de todos os outros drives importados. Ainda, o novo boot checa a memória no micro, verifica sua capacidade (nunca se sabe... estamos no Brasil), e, caso sua disponibilidade não seja suficiente, envia uma mensagem de erro para tela, entrando em seguida num loop infinito. Caso contrário, aquele seu joguinho favorito que não funcionava, agora vai estar aos seus comandos.

É importante observar que este programa não trará solução para todos os boot's existentes no mercado, mas sua rotina principal (ver listagem em assembler) poderá servir de base para codificação de outros boot's que necessitam de "algo mais", como é o caso do ARKANOID da Taito, que precisa inibir o hook HTIME para funcionar.

Especificações técnicas do SSC:

- O SSC, para evitar danos ao programa, checa o identificador de arquivos binários e o identificador de cartuchos.
- Os programas de 16k só funcionarão se tiverem seu endereço original assinalado em &H4000.
- Todos os programas processados pelo SSC, terão os endereços assinalados em &H9000 - &HD075 - &HD000 (inicial-final-execução).
- Rotina STOP MAPPER DRIVE.
- Rotina de avaliação de erros.
- Funciona 90% nos programas de 16k/32k, da Konami, e 60% nos programas de 16k/32k, de outras marcas.

MSXDEBUG E SCREEN IV

Os Programas que faltavam na sua biblioteca

MSXDEBUG

DISP endereço
Mostra o conteúdo da memória dado por endereço. Para avançar ou retroceder, utilize as teclas dos cursores. Estando na posição desejada, pressione «CR». Neste momento, você poderá alterar o conteúdo desta memória ou se deslocar ao longo dos códigos livremente. Para alterar utilizando diretamente o teclado, pressione «CR» novamente. Repita sempre que precisar mudar o modo de edição. Para sair, basta pressionar «ESC».

EXEC endereço
Executa este endereço sem break-points.

MOVE início, fim, destino
Move um bloco na memória. Início deve ser menor que o fim.

FILL início, fim, dado
Preenche uma área de memória especificada por início e fim.

DIR
Mostra o diretório do disco de forma simples.

DSAVE nome. ext, início, fim
Salva bloco limitado por início e fim no disco com o nome nome. ext. Salva qualquer coisa em qualquer lugar.

DLOAD nome. ext. endereço
Carrega qualquer arquivo, desde que esteja no disco e o coloca no endereço dado por endereço. Nestes três comandos não é possível especificar o drive. Será utilizado aquele que estava habilitado antes de executar o MSXDEBUG.

DOS
Encerra as atividades no MSXDEBUG e retorna ao DOS.

SOMA
Soma os valores entre início e fim e apresenta o resultado em HEXADECIMAL.

BUSCA
Procura determinada seqüência de caracteres.

BSAVE
Salva um arquivo em disco no formato do Basic.

BLOAD
Carrega um arquivo no formato Basic do disco para a memória.

APPEND
Junta um bloco de memória.

DASS
Traduz os códigos em linguagem de máquina para os Mnemônicos do Z-80.

SCREEN IV

SCREEN 4	: Habilita tela de 64 x 24 linhas
FILL	: Um paint 8 vezes mais rápido
OVER	: Sobrepõe textos
CLS ON/OFF	: Inibe e habilita o CLS quando se muda o número de colunas com o comando WIDTH
WIDTH	: Alterado para comportar 64 colunas
SCREEN SAVE/LOAD	: Grava e carrega telas do cassete
SCREEN OUT	: Imprime telas na impressora
LIST	: Apresenta a listagem de um programa, sendo que a mesma é interrompida quando a tela fica cheia, pedindo que se pressione uma tecla pra prosseguir.

REVENDEDORES:
MSX TRONIC INFORMÁTICA LTDA. R. Senador Vergueiro 207/1205 - Flamengo - 22230 - Rio de Janeiro - RJ - Tel.: (021) 552-4581 • **GAME OFF TIME SOFTWARE E HARDWARE** Av. Jabaquara 1598 sala 08 - Saúde - 04046 - São Paulo - SP - Tel.: (011) 581-2739 • **MANÍACOS DO MSX** Av. Jabaquara 99 sala 54/5º andar - Vila Mariana - 04050 - São Paulo - SP - Tel.: (011) 276-7665 • **CHAMPION SOFTWARE LTDA.** R. Clélia 1837 - Lapa - 05042 - São Paulo - SP - Tel.: (011) 65-2030 • **ECTRON ELETRÔNICA LTDA.** R. Dr. César 131 - Santana - 02089 - São Paulo - SP - Tel.: (011) 290-7266 • **TOYGAMES INFORMÁTICA LTDA.** R. Galvão Bueno 714 - Conj. 16 - Liberdade - 01051 - São Paulo - SP - Tel.: (011) 277-4878 • **FARAH'S INFORMÁTICA COMP. E SIST. LTDA.** R. São Bento 365 S/Loja - Centro - 01011 - São Paulo - SP - Tel.: (011) 37-3437 • **A&A SOFTWARE** Caixa Posral 201 - Sapiranga - RS • **TOP DATA INF. TEC. LTDA.** Av. Washington Soares 85 Lj 106/113 - Edson Queiroz - 60810 - Fortaleza - CE - Tel.: (085) 239-2798 • **SUN PHOTO INFORMÁTICA LTDA.** R. Torres Câmara 440 - Aldeota - 60150 - Fortaleza - CE - Tel.: (085) 244-2308 • **CASA DA INFORMÁTICA LTDA.** R. Visconde de Pirajá 281 Sobreloja 203 - Ipanema - 22410 - Rio de Janeiro - RJ - Tel.: (021) 521-2844

```
1000 ; SOFTWARE SIMULADOR DE CARTUCHOS 16k - SSC16
1010 ; DE 4000H Ate 7FFFH
1020 ; autor: Carlyle Zamith
1030 ;
1040 ;------------------------------------------labels
1050 CHPUT:   EQU    000A2H      ;print no video
1060 CALSLT:  EQU    0001CH      ;inter-slot
1070 INITXT:  EQU    0006CH      ;inicializa screen 0
1080 ENASLT:  EQU    00024H      ;enable slot
1090 EXPTBL:  EQU    0FCC1H      ;slot expandido?
1100 PROCNA:  EQU    0FD89H      ;nome do call
1110 FORCLR:  EQU    0F3E9H      ;cor de frente
1120 BAKCLR:  EQU    0F3EAH      ;cor de fundo
1130 BDRCLR:  EQU    0F3EBH      ;cor de borda
1140 LINL40:  EQU    0F3AEH      ;largura da tela
1150          ORG    0D000H      ;endereco inicial
1160 ;
1170 ;------------------------------------------inicio
1180                 DI
1190          LD     IY,(0F347H)  ;slot interface
1200          LD     IX,4029H     ;rotina stop-drive
1210          CALL   CALSLT       ;executa
1220          LD     A,080H
1230 NEXT1:   PUSH   AF           ;salva endereco
1240          CALL   RAM          ;goto ram
1250          POP    AF           ;restaura endereco
1260          LD     B,A
1270          LD     A,C
1280          AND    A
1290          JP     M,TRANSF
1300          LD     A,B
1310          INC    A
1320          CP     90H          ;mais slots
1330          JR     C,NEXT1
1340          CALL   SLOT
1350          LD     A,01H
1360          LD     (FORCLR),A
1370          LD     A,0FH
1380          LD     (BAKCLR),A
1390          LD     (BDRCLR),A
1400          LD     A,25H
1410          LD     (LINL40),A
1420          CALL   INITXT
1430          LD     HL,MSGERR
1440 PRINT:   LD     A,(HL)
1450          AND    A
1460 LOOP:    JR     Z,LOOP
1470          CALL   CHPUT
1480          INC    HL
1490          JR     PRINT
1500 ;
1510 ;------------------------------------------prc_ram
1520 RAM:     DI
1530          LD     HL,5FFDH     ;total de slots
1540          PUSH   BC           ;salva slot counter e pag.
1550          PUSH   HL           ;salva ponteiro do EXPTBL
1560          CALL   ENASLT
1570          POP    HL           ;restaura ponteiro EXPTBL
1580          POP    BC           ;restaura slot e pag.
1590          LD     A,(HL)
1600          CPL
1610          LD     (HL),A
1620          CP     (HL)
1630          JR     NZ,NEXT2
1640          SET    7,C
1650 NEXT2:   CPL
1660          LD     (HL),A
1670          RET
1680 ;
1690 ;------------------------------------------select slot
1700 SLOT:    LD     A,EXPTBL
170           LD     HL,04000H
1720          CALL   ENASLT
1730          EI
1740          RET
1750 ;
1760 ;------------------------------------------transferencia
1770 TRANSF:  LD     A,B
1780          LD     (PROCNA),A
1790          LD     HL,08200H
1800          LD     DE,04000H
1810          LD     BC,02000H
1820          LDIR
1830          LD     HL,0A300H
1840          LD     DE,06000H
1850          LD     BC,02000H
1860          LDIR
1870          JP     04004H
1880 ;
1890 ;------------------------------------------mensagem
1900 MSGERR:  DEFM 'Este computador MSX '
1910          DEFM 'nao tem memoria'
1920          DEFB 0DH,0AH
1930          DEFM 'suficiente para '
1940          DEFM 'executar este programa.'
1950          DEFB 0DH,0AH,0DH,0AH
1960          DEFM 'Press RESET '
1970          DEFM 'para continuar.'
1980          DEFB 00H
1990          END
```

```
1000 ' SSC - Programa desenvolvido
1010 ' especialmente para Revista CPU
1020 '
1030 CLEAR 200,&H8FFF:T$=CHR$(1)+CHR$(64+23)
1040 COLOR 15,1:SCREEN0:WIDTH 40:KEYOFF
1050 PRINT T$;T$;" Soft Simulador de Cartucho 16k/32k ";T$;T$
1060 LOCATE 0,3:LINEINPUT "TAMANHO 16k/32k: ";TA$
1070 IF VAL(TA$)<>16 AND VAL(TA$)<>32 THEN BEEP:LOCATE ,CSRLIN-1:GOTO 1060
1080 IF VAL(TA$)=16 THEN TA=1 ELSE TA=2
1090 FOR Z=1 TO TA:LOCATE 0,3+Z
1100 PRINT "NOME DO PROGRAMA #";RIGHT$(STR$(10+Z),1);": ";:LINEINPUT SO$(Z)
1110 NEXT Z:PRINT:PRINT "Espere um momento ..."
1120 ON ERROR GOTO 1480
1130 Z=1:GOSUB 1310
1140 GOSUB 1590
1150 IF TA=2 THEN GOSUB 1460
1160 BSAVE SO$(Z),&H9000,&HD100,&HD000
1170 IF TA=1 THEN 1210
1180 Z=2:GOSUB 1310
1190 GOSUB 1750
1200 BSAVE SO$(Z),&H9000,&HD100,&HD000
1210 PRINT:PRINT "Operação completada com sucesso."
1220 PRINT:PRINT "Você deseja continuar (S/N)? ";:A$=INPUT$(1)
1230 IF A$<>"S" AND A$<>"N" THEN BEEP:LOCATE ,CSRLIN-1:GOTO 1220
1240 IF A$="S" THEN RUN
1750 CLS:PRINT "SSC16K  -  Soft Simulador de Cartucho"
1260 FOR I=1 TO 37:PRINT T$;:NEXT:PRINT
1270 PRINT "Copyright C(1990) DATASET INFORMATICA"
1280 PRINT:PRINT:PRINT "AUTORES: Carlyle Zamith &"
1290 PRINT SPC(9)"Arnaldo Marques"
1300 NEW
1310 OPEN SO$(Z) FOR INPUT AS#1
1320 B1$=INPUT$(1,#1)
1330 IF HEX$(ASC(B1$))<>"FE" THEN ERROR 255
1340 X$=INPUT$(6,#1):ID$=INPUT$(2,#1):CLOSE
1350 IF TA=1 THEN IF ID$<>"AB" THEN ERROR 253
1360 FOR I=1 TO 6
1370 M(I)=ASC(MID$(X$,I,10))
1380 NEXT
1390 EI=M(2)*256+M(1)
1400 EF=M(4)*256+M(3)
1410 EX=M(6)*256+M(5)
1420 M(0)=(EF-EI+1)
1430 IF INT(M(0)/1000)<>16 THEN ERROR 254
1440 BLOAD SO$(Z),36864!-EI
1450 RETURN
1460 POKE &HD072,&HC3:POKE &HD073,&H59
1470 POKE &HD074,&HD0:POKE &HD075,&H0:RETURN
1480 BEEP:LOCATE ,CSRLIN-1
1490 IF ERR=53 THEN PRINT "ERRO! EU NÃO ENCONTREI O ARQUIVO.":RESUME 1570
1500 IF ERR=56 THEN PRINT "ERRO! NOME DO PROG. FORA DO PADRAO.":RESUME 1570
1510 IF ERR=68 THEN PRINT "ERRO! O DISCO ESTA PROTEGIDO.":RESUME 1570
1520 IF ERR=70 THEN PRINT "ERRO! EU NÃO CONSIGO LER O DISCO.":RESUME 1570
1530 IF ERR=253 THEN PRINT "ERRO! SEM IDENTIFICADORES DE CARTUCHO.":RESUME 1570
1540 IF ERR=254 THEN PRINT "ERRO! O PROGRAMA É DIFERENTE DE 16K.":RESUME 1570
1550 IF ERR=255 THEN PRINT "ERRO! NÃO ESTA EM CÓDIGO BINARIO.":RESUME 1570
1560 PRINT "ERRO! CÓDIGO Nº ";ERR;" -VERIFIQUE O MANUAL DO MICRO.":RESUME 1570
1570 TIME=0
1580 IF INT(TIME/60)<5 THEN 1580 ELSE RUN
1590 RESTORE 1600:FOR I%=0 TO 226:READ A$:POKE &HD000+I%,VAL("&H"+A$):NEXT:RETURN
1600 DATA F3,FD,2A,47,F3,DD,21,29,40,CD,1C,00,3E,80,F5,CD
1610 DATA 44,D0,F1,47,79,A7,FA,63,D0,78,3C,FE,90,38,EF,CD
1620 DATA 59,D0,3E,01,32,E9,F3,3E,0F,32,EA,F3,32,EB,F3,3E
1630 DATA 25,32,AE,F3,CD,6C,00,21,76,D0,7E,A7,28,FE,CD,A2
1640 DATA 00,23,18,F6,21,FD,5F,C5,E5,CD,24,00,E1,C1,7E,2F
1650 DATA 77,BE,20,02,CB,F9,2F,77,C9,3A,C1,FC,21,00,40,CD
1660 DATA 24,00,C9,78,32,89,FD,21,00,90,11,00,40,01,00,40
1670 DATA ED,B0,2A,02,40,E9,45,73,74,65,20,63,6F,6D,70,75
1680 DATA 74,61,64,6F,72,20,4D,53,58,20,6E,61,6F,20,74,65
1690 DATA 6D,20,6D,65,6D,6F,72,69,61,0D,0A,73,75,66,69,63
1700 DATA 69,65,6E,74,65,20,70,61,72,61,20,65,78,65,63,75
1710 DATA 74,61,72,20,65,73,74,65,20,70,72,6F,67,72,61,6D
1720 DATA 61,2E,0D,0A,0D,0A,50,72,65,73,73,20,52,45,53,45
1730 DATA 54,20,70,61,72,61,20,63,6E,6E,74,69,6E,75,61,72
1740 DATA 2E,00,00
1750 RESTORE 1760:FOR I%=0 TO 50:READ A$:POKE &HD000+I%,VAL("&H"+A$):NEXT:RETURN
1760 DATA F3,21,FD,7F,1E,3F,3A,22,FB,CD,14,00,3A,89,FD,21
1770 DATA 00,40,CD,24,00,21,00,90,11,00,80,01,00,40,ED,B0
1780 DATA FD,2A,47,F3,DD,21,29,40,CD,1C,00,21,02,40,5E,23
1790 DATA 56,D5,C9
```

MOZART: UM COMPILADOR BASIC DE NÍVEL PROFISSIONAL

Paulo Roberto Pinheiro Elias e
Sergio Guy Pinheiro Elias

Acabamos de receber o mais novo lançamento da CIBERTRON ELETRÔNICA, produtora de programas conhecidos no mercado de MSX, entre os quais, um dos mais usados Processadores de Texto, o MSX-WORD. Trata-se desta vez do Compilador BASIC denominado (por razões ainda não explicadas) de MOZART. Gostaríamos de passar aos leitores de CPU nossas impressões a seu respeito.

O MOZART difere radicalmente de outros compiladores de MSX BASIC, pelo fato de produzir um programa-objeto em linguagem de máquina, executável em ambiente DOS, ao contrário dos outros compiladores pré-existentes, que obrigavam o usuário a carregá-los para o micro, no momento em que o programa compilado deveria ser rodado. Na realidade, muitos "compiladores" disponíveis no mercado apenas possibilitavam carregar o programa em BASIC como se fosse um programa em linguagem de máquina (comando BLOAD), mantendo a sua listagem, em alguns casos, bloqueada.

COMO FUNCIONA UM COMPILADOR COMO O MOZART:

Para aqueles que não estão familiarizados com o "computês", torna-se necessário explicar alguns termos e conceitos relacionados com a compilação de um programa:

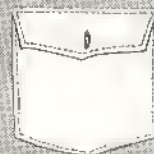
Com exceção da linguagem de máquina (conjunto de instruções codificadas por sequências de zeros e uns), todas as linguagens de programação, para serem entendidas pelo computador, precisam ser convertidas para códigos de máquina, de maneira a que os microprocessadores que recebem estas instruções possam saber o que fazer com elas.

Esta conversão é semelhante à tradução de um idioma estrangeiro quando alguém de fora nos visita. Vamos supor que este visitante vem passear em nossa terra, mas não entendemos nada do que ele fala. Assim, chamamos um *intérprete* para fazer a tradução *simultânea* do que o visitante tenta nos dizer. Quanto mais perfeita for esta tradução, mais claramente entenderemos os seus desejos e mais corretamente o atenderemos. Ao sair de nosso país, o visitante deixa uma carta de agradecimento, mas, como a mesma está escrita em sua língua, chamamos um tradutor que irá reescrever aquele documento em *nossa* língua. Quando tivermos esta tradução em mãos, poderemos lê-la rapidamente, pois estaremos olhando para palavras previamente traduzidas, ao contrário do que acontecia antes, quando o intérprete demorava um certo tempo para traduzir o que estava sendo falado num dado momento.

O BASIC do MSX, para ser entendido pelo micro, precisa ser *interpretado* no momento da execução do programa, o que acarreta uma certa perda de tempo nesta última tarefa, visto que, a cada instrução encontrada, o interpretador BASIC vai buscar a rotina em linguagem de máquina a ela pertinente. Deste modo, é de se esperar que um programa em BASIC seja mais lento ao ser executado do que um programa em linguagem de máquina.

Se, entretanto, este programa puder ter *previamente* seus comandos traduzidos para as rotinas em códigos de máquina, a sua velocidade de execução aumentará significativamente. Este processo denomina-se COMPILAÇÃO referente ao ato de pesquisar e incluir, no corpo do programa, somente as rotinas ligadas à sua execução. Um diagrama comparativo entre interpretação e compilação está mostrado na Figura 1.

Assim, podemos dizer que um Compilador é um *tradutor* das instruções de uma linguagem de programação para os seus respectivos códigos de máquina. A qualidade de um compilador é diretamente proporcional à sua capacidade de traduzir corretamente todos os comandos incluídos no programa. Caso existam instruções que ele não entenda, a sua tradução não será possível. Por outro lado, como todas as instruções de programação envolvem uma série de tarefas ou *ROTINAS* a serem executadas, quanto mais clara for esta tradução, mais fielmente estas rotinas serão executadas.

A tradução, por seu turno, é dependente de certas regras, que poderão ajudar a tarefa de compilação. Uma delas se refere à *legibilidade* do texto (programa-fonte) a ser traduzido. É geralmente necessário prestar bastante atenção na SINTAXE dos comandos e instruções, sob pena de não permitir ao compilador compreender corretamente o que se está pedindo. Ligada a esta necessidade, está

também a tarefa de facilitar a tradução, modificando a sintaxe de certas instruções, de modo a que as mesmas passem a ser entendidas e atinjam, na prática, o seu real significado.

MODO DE OPERAÇÃO DO MOZART:

Quem alguma vez já manipulou um compilador em ambiente DOS, do tipo TURBO PASCAL, da Borland, ou MACRO-80, da Microsoft, não sentirá dificuldade em utilizar este programa. Na realidade, verifica-se, logo aos primeiros contatos com seu modo de operação e comandos, que o MOZART possui os mesmos "switches" (chaves que são adicionadas à linha de comando) e instruções que o M80 acima citado.

Diríamos até que, frente a uma certa exigüidade do Manual de Instruções que acompanha o produto, levará vantagem aquele usuário que porventura possuir o Manual original da Microsoft sobre o M80.

O MOZART vem contido em apenas um disquete, mas acompanhado de um soquete de joystick, chamado pelo Manual de "chave de segurança" e que deverá ser conectado à entrada de joystick "B". Sem esta chave, a gravação do arquivo relocável produzido pelo compilador não é efetuada. Assim, o software torna-se protegido contra a famigerada e injustificada pirataria que caracteriza o padrão MSX no país, embora os programas contidos no disquete estejam totalmente liberados para cópia de segurança.

Da mesma forma que o M80, o MOZART produz inicialmente um arquivo relocável, com extensão REL, a partir do programa-fonte em BASIC no formato ASCII, o qual poderá ser facilmente obtido através de um Processador de Texto ou, no caso de um programa pré-existente, através do comando SAVE "<programa>", A constante do BASIC. Qualquer Editor de Textos que produza arquivos não formatados (sem códigos de controle além do texto) poderá ser usado. A CIBERTRON, por motivos óbvios, sugere o uso do MSX-WORD da Versão 3.0, mas tal procedimento não é prático, pois este programa não opera em ambiente DOS.

Este arquivo relocável ainda não é o programa compilado, pois a ele devem ser adicionadas as rotinas existentes nas bibliotecas disponíveis ou naquelas criadas pelo próprio usuário. A obtenção deste arquivo facilita certas tarefas do compilador, entre as quais, a adição de alguma rotina escrita em outra linguagem de programação, necessária à execução do programa compilado. No próprio Manual, o usuário recebe um exemplo de seu interesse, relativo à adaptação de comandos tipo USR e DEFUSR, através de rotinas em linguagem de máquina.

Depois que o arquivo relocável é criado, ele se torna objeto de uso para o "Linker" ("lincador" ou "ligador") fornecido junto com o MOZART. Ele é denominado MLK (provavelmente abreviação de Mozart Linker), e é bastante assemelhado ao LINK-80 (ou L80), da Microsoft, utilizando comandos e "switches" deste último, obtendo os mesmos resultados. O Manual do usuário, neste ponto, é muito fraco, pois ele apresenta apenas os "switches" /N (que produz um arquivo contendo o programa-fonte transformado em programa-objeto, com extensão –COM) e /E (que propicia a saída do MLK e retorno ao DOS).

Além de não apresentar os outros "switches" o Manual também não explica quais os números obtidos após a "lincagem". No caso, os mesmos se referem à ação de /N, que faz o MLK imprimir o endereço inicial, o endereço do próximo byte disponível e o número de páginas de 256 Bytes utilizadas pelo Linker. Talvez o Manual não dê muita ênfase a estes dados pelo fato dos mesmos não representarem informações importantes para um grande número de usuários. Como os programas-fonte estão escritos em BASIC e rodarão em ambiente DOS, o primeiro destes valores tende a ser 0103H em todos os casos onde a compilação não é feita com relocação do endereço inicial.

Alguns "switches" não mencionados pelo Manual são particularmente úteis para alguns tipos de usuários, como por exemplo, /G (que roda automaticamente o programa carregado pelo Linker), /M (que lista todas as referências globais definidas ou não nas bibliotecas consultadas) ou ainda /R (que provoca um "reset" no programa, possibilitando carregar outros arquivos). Cremos que seria interessante que a CIBERTRON publicasse e distribuísse aos usuários do MOZART um adendo contendo maiores informações sobre a "lincagem", de modo que o MLK pudesse ser melhor empregado.

Outra informação que o Manual omite, embora facilmente identificável pelo operador, é a existência de um "prompt" na forma de um asterisco ("*"), tanto no MOZART quanto no MLK. Isto, na prática, significa a possibilidade de se entrar com ambos os programas em operação pela simples digitação do nome dos respectivos arquivos na linha de comando do DOS e posteriormente digitar linhas de comando diversas dentro dos programas, dispensando assim acioná-los repetidas vezes.

A "lincagem" produz, como foi comentado, um arquivo-objeto com extensão default COM, que caracteriza os programas compilados executáveis, carregáveis pelo Sistema Operacional DOS. Em tese, seria possível criar programas-fonte com tamanho maior do que os costumeiros 24 KBytes livres no ambiente de BASIC de DISCO existentes no MSX da série 1.0, em sua configuração habitual. No tempo em que se costumava operar o micro exclusivamente com fita cassete, era normal carregar arquivos em BASIC com cerca de 26 ou mais Kbytes, o que agora pode passar a ser novamente factível.

COLOCANDO O MOZART EM OPERAÇÃO:

O MOZART opera normalmente até a etapa de gravação do programa-objeto em um arquivo relocável, fase na qual o programa verifica se a chave de segurança fornecida junto com o "software" está instalada na entrada "B" dos joysticks. Se isto não for verdade, o programa emite mensagens de advertência ao usuário, ao mesmo tempo em que bloqueia completamente o computador, obrigando a uma nova partida a frio. No caso de um dos autores deste artigo, foi detectado um problema na conexão de joystick, o qual impediu que o programa terminasse de executar suas tarefas. Mesmo depois da verificação dos contatos, o defeito persistiu e demonstrou serem críticas as condições de funcionamento quando há qualquer obstrução nos circuitos do micro. Se por um lado, é justo que o fabricante se proteja, por outro, levando em consideração a má qualidade dos equipamentos nacionais, este tipo de proteção promete grandes dores de cabeça tanto ao fabricante quanto usuário.

O "Linker' MLK trabalha sem qualquer tipo de proteção e o seu funcionamento é bastante lento. Esta lentidão é diretamente proporcional ao número de tarefas que ele é obrigado a executar pelo emprego dos diversos "switches" digitados na linha de comando. Algumas mensagens que aparecem na tela, durante a execução do programa, ficam corretamente dispostas em vídeo de 80

colunas, o que confere um ar "profissional" ao seu funcionamento.

O MOZART é bastante suscetível a não entender palavras truncadas, que, infelizmente, é a forma de digitação preferida dos programadores profissionais e hobbyistas, para economizar espaço na memória. Assim, muitos comandos acusarão "erros de sintaxe" e deverão ser redigitados. Possivelmente, muitas linhas de programa terão que ser rearranjadas, de modo a evitar a saturação de instruções por linha.

Uma boa iniciativa da CIBERTRON foi a de incluir um programa DEMO, com o seu respectivo fonte, que o usuário poderá inspecionar através de um processador de textos. Além disso, será possível comparar o funcionamento deste programa com e sem o artifício da compilação, quando então ficarão notórias as vantagens que este processo traz ao usuário, principalmente com relação à rapidez de execução.

Figura 1 – Comparação de execução de um programa em BASIC com e sem o artifício da compilação. Note que o programa compilado, ao ser executado, não passa pelas mesmas etapas de um programa interpretado, já que a tradução dos comandos e a pesquisa das rotinas necessárias à sua execução é feita previamente.

AVALIAÇÕES FINAIS:

Sem dúvida, este é um dos melhores lançamentos feitos nos últimos tempos, dentro da linha MSX. Na medida em que se abre portas de escape aos programadores que usam este micro de forma conseqüente, fica mais tolerável aturar as inevitáveis faltas de recursos que caracterizam a série presentemente comercializada no Brasil, diga-se de passagem, fora de linha, segundo consta, desde 1985 nos países do primeiro mundo.

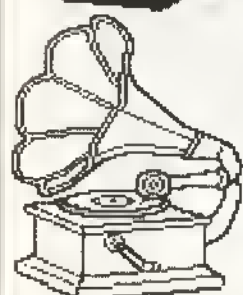
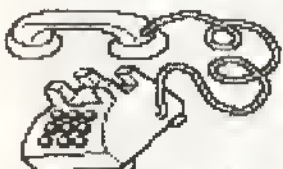
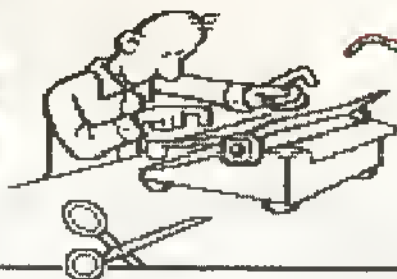
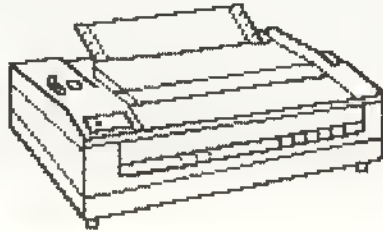
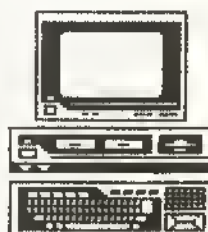
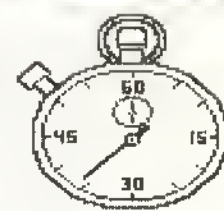
Especialmente útil, este compilador permite, para aqueles que se interessam, o aproveitamento da linguagem BASIC constante no MSX, na quase totalidade de seus comandos. Levando-se em consideração os vários recursos, inclusive gráficos, desta linguagem, e a rapidez que este compilador propicia, podemos prever um maior estímulo no seu emprego pelos usuários que não desejam se envolver com linguagens de nível mais baixo, como o ASSEMBLY.

Além disso, o compilador MOZART traz uma série de vantagens, entre elas, a alocação dinâmica de memória na área de variáveis, facilitando sobremaneira a redação e a execução daqueles programas que dependem deste recurso.

Um outro aspecto importante para muitos programadores é a proteção dos programas-fonte, impedindo a leitura das rotinas proprietárias, muitas vezes criadas com muito trabalho e dedicação.

Concluindo, diríamos que, apesar dos transtornos iniciais que a adaptação dos programas-fonte certamente causará, o uso deste compilador será de grande importância para todos aqueles que desejam melhorar a performance de seus programas em BASIC, sem precisar trocar de computador, mesmo porque, enquanto permanecer neste país esta infame lei de reserva de mercado na área dos microcomputadores, este certamente será o MSX com que haveremos de conviver por longos e longos anos, amém !...

AGRADECIMENTOS:

Desejamos agradecer a Leonardo Senna e ao pessoal da CIBERTRON, pelo envio do compilador MOZART, e aproveitamos o ensejo, não só para parabenizá-los pela coragem deste lançamento (mostrando que a empresa tem uma mentalidade realmente progressista), como também augurar os nossos melhores votos para que, desta vez, a CIBERTRON não sofra mais a dilapidação de seus investimentos pelos agentes sem escrúpulo que vem comercializando seus programas, na calada da noite, a preços aviltantes.

PAGE MAKER
O primeiro, mais simples e mais econômico programa de DESK-TOP PUBLISHING criado para o sistema MSX. Ideal para quem quer tirar o máximo proveito da sua impressora gráfica em diplomas, trabalhos escolares, cartazes, avisos, estórias em quadrinhos, tabelas, e muito mais ...
Em disco com manual por Cr$ 600,00
MSX PAGE MAKER KIT
FONTES #1 (LETRAS).......Cr$ 250,00
FONTES #2 (LETRAS).......Cr$ 250,00
FONTES #3 (LETRAS).......Cr$ 250,00
FONTES #4 (LETRAS).......Cr$ 250,00
CARTOONS #1 (FIGURAS)...Cr$ 250,00
CARTOONS #2 (FIGURAS)...Cr$ 250,00
TITLES #1 (TITULOS).....Cr$ 250,00
SQUARES #1 (MOLDURAS)...Cr$ 250,00
KIT COMPLETO - APENAS Cr$ 2.000,00
MSX PAGE MAKER "CLIP-ART-KIT"
Uma coleção de figuras inéditas em formato "SHAPE" para utilização no MSX PAGE MAKER ou seus compatíveis
4 discos por apenas - Cr$ 1.000,00
TOPCAD
Um sensacional "EDITOR DE PROJETOS PROFISSIONAIS" para seu MSX. Ideal para desenho de circuitos eletrônicos, plantas de arquitetura, fluxogramas, hidraulica, elétrica, etc.
Em disco com manual por Cr$ 800,00
NEMESIS
ENVIE VALE POSTAL OU CHEQUE NOMINAL A NEMESIS INFORMATICA LTDA NO ENDEREÇO:
CAIXA POSTAL 4583/20001
RIO DE JANEIRO - RJ.
OU VENHA PESSOALMENTE AO NOSSO "SHOW-ROOM":
RUA SETE DE SETEMBRO 92
SALA 1910 - CENTRO
RIO DE JANEIRO - RJ.
SUPER JOGOS PARA MSX!
RAMBO III finalmente em disco por - Cr$ 250,00!
"CHASE HQ" e MOTORCYCLE MADNESS por Cr$ 300,00!
60
15
30
45

LISTAGEM PAGINADA DE PROGRAMAS FONTE EM FORMATO ASCII

Sergio Guy Pinheiro Elias e
Paulo Roberto Pinheiro Elias

INTRODUÇÃO:

Freqüentemente o usuário que se inicia em programação sente falta, com o correr do tempo, de recursos que possam auxiliá-lo na tarefa de programar. Na medida em que os programas aumentam em tamanho, aumenta proporcionalmente a dificuldade de ler o que está sendo digitado, pois parte do texto vai ficando para trás, impecilhando o usuário na hora de conferir alguma coisa de seu interesse, disposta em linhas anteriores. Por este motivo, é comum a adoção, pelas softhouses, de editores de texto na elaboração de programas em qualquer tipo de linguagem.

Pelo mesmo motivo, alguns compiladores existentes no comércio já vem com seus próprios editores de texto, por vezes com vantagens, como é o caso do Turbo Pascal, cujo editor está conjugado ao depurador ("debugger") de compilação. No momento em que esta se realiza, toda vez que um erro é detectado, o editor contendo o programa fonte é acionado e o erro automaticamente posicionado para o usuário, facilitando a redigitação corretiva.

USANDO EDITORES DE TEXTO NO MSX PARA REDIGIR PROGRAMAS FONTE:

Praticamente todos os editores de texto para os micros MSX são capazes de permitir a digitação de um programa fonte, mesmo que a linguagem utilizada seja o próprio BASIC residente na máquina. As raras exceções são as Versões do MSX-WORD anteriores à 3.0 ou qualquer uma das adaptações do antigo TASWORD em fita cassete. Em ambiente DOS, ideal para confecção de certos tipos de programas, como por exemplo, o dBASE, os dois editores de texto que mais se destacam são, a nosso ver, o MED (MICROSOFT FULL SCREEN EDITOR) ou o SCED (seu parente mais antigo e limitado), e o WordStar. Este último permite rodar qualquer programa sob o DOS, que tenha extensão. COM, recurso este que tem algumas vantagens:

No caso de programas em linguagem dBASE, por exemplo, o usuário poderá inicializar o WordStar, usando inicialmente a opção N (arquivos Não documento) para processar o programa em linguagem dBASE. Depois de terminada a digitação, entra-se na opção R para rodar o dBASE e assim testar se o programa funciona como desejado. Quando o usuário encerra a sessão dBASE com a instrução QUIT, o WordStar reassume as suas operações, permitindo reeditar o programa se for necessário. O único cuidado a ser tomado é que ambos os programas deverão estar contidos no mesmo disquete, alojado no drive A. Fora isso, poder-se-á ir e vir por este caminho. A relativa lentidão que este método apresenta, se comparado com o acesso ao editor do próprio dBASE. é amplamente compensada com as facilidades e recursos de edição e impressão do WS, amplamente superiores neste particular. Ao contrário do que muitos autores afirmam, o editor de textos do dBASE é bem deficiente e contra-indicado para a redação de programas fonte (principalmente dos extensos), mesmo nas suas Versões do IBM-PC. Evidentemente, que programas muito pequenos deverão ser escritos diretamente no editor dBASE, para poupar o usuário desta mão de obra de ir e vir de um software a outro.

USANDO A IMPRESSORA PARA AUXILIAR A PROGRAMAÇÃO:

Além de processadores de texto, uma das mais úteis ferramentas de trabalho para os programadores são as impressoras. Ao obter uma listagem do programa, é possível avaliar com mais rapidez a sua estrutura e a sua lógica, além de se poder fazer anotações à margem do formulário, que poderão ajudar o programador na descoberta de erros ou na elaboração de novos melhoramentos.

Em princípio, os editores de texto podem se prestar para esta listagem, mas em alguns casos, o filtro de impressão que vem embutido no programa poderá interpretar algum caracter de modo a alterar a listagem original.

Uma outra possibilidade é oferecida no programa da Figura 1. A idéia aqui é aproveitar a mesma estrutura dos programas constantes de nosso artigo sobre "Alternativas de impressão para o MSX-WORD", de tal forma que aqueles que digitaram algum daqueles programas, poderão aproveitá-los quase que integralmente.

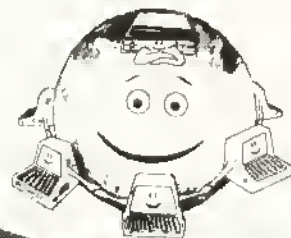
Neste programa, alguns parâmetros de impressão foram propositalmente predefinidos para a listagem pretendida e dificilmente deverão ser mudados. Neste caso está o offset de impressão, que deve ser zero quando se deseja aproveitar integralmente a extensão do formulário. Numa situação menos radical está o estabelecimento do número de colunas (ou caracteres/linha), para o qual pode-se aumentar o seu valor, de acordo com formulários mais largos ou ainda de acordo com o "pitch" das letras da impressora (por exemplo, no modo condensado).

Outros recursos como cabeçalho, rodapé e principalmente a numeração ou não das páginas, estarão à disposição do usuário, no Menu Principal. O programa lê diretamente no disco o programa fonte, pela instrução LINE INPUT#, como explicado no artigo anterior. A limitação de 255 Bytes/linha desta instrução não deverá , em princípio, constituir problema. Com este programa, pode-se listar qualquer programa gravado no formato ASCII. No caso de programas em BASIC redigidos da forma convencional no MSX, estes deverão ser gravados com o argumento ",A" na linha de comando.

```
10 'PROGRAMA PARA IMPRIMIR
20 'PROGRAMAS-FONTE GRAVADOS EM FORMATO ASCII
30 'AUTORES:S.G.P.E./P.R.P.E.
40 'DATA: NOVEMBRO DE 1989
50 KEY OFF:LOCATE,,0:CLEAR 1000
60 ON ERROR GOTO 210
70 A$="NÃO":B$="NÃO":A=1:B=80:C=55:F1=0:F2=1:L$="SIM":R=0
80 CLS
90 PRINT"IMPRIME PROGRAMAS-FONTE (FORMATO ASCII)":LOCATE13,2:PRI
NT"Menu Principal":LOCATE1,4:PRINT"(  ) indicam estado atual das
 funções"
100 LOCATE,6:PRINT"1 - IMPRIME CABEÇALHO (";A$;")"
110 LOCATE,8:PRINT"2 - IMPRIME RODAPÉ (";B$;")"
120 LOCATE,10:PRINT"3 - NUMERAÇÃO DAS PAGS. (";L$;")
130 LOCATE,11:PRINT"    REDEFINIÇÃO DA NUMERAÇÃO (";R;")"
140 LOCATE,13:PRINT"4 - DEFINE Nº DE CARACTERES/LINHA (";B;")"
150 LOCATE,15:PRINT"5 - DEFINE Nº DE LINHAS/PÁGINA (";C;")"
160 LOCATE,17:PRINT"6 - IMPRIME PROGRAMA FONTE"
170 LOCATE,19:PRINT"7 - ABANDONA O PROGRAMA"
180 LOCATE,21:PRINT"Sua opção: ";
190 C$=INPUT$(1):IF C$<"1" OR C$>"7" THEN 80 ELSE PRINT C$:FOR T
=1 TO 100:NEXT T
200 ON VAL(C$) GOTO 240,300,380,440,490,540,930
210 IF ERR<>53 THEN CLS:PRINT"FOI DETECTADO UM ERRO Nº:";ERR;"LI
NHA:";ERL:GOTO 230
220 IF ERL=590 THEN PRINT:PRINT"Arquivo não existe":CLOSE:PRINT:
PRINT"Tecle algo":T$=INPUT$(1):RESUME 540
230 CLOSE:PRINT:PRINT"Tecle algo":T$=INPUT$(1):RESUME 80
240 CLS:PRINT"OPÇÃO PARA IMPRESSÃO DE CABEÇALHO"
250 LOCATE,5:PRINT"Confirma impressão de cabeçalho (S/N) ?";:C$=
INPUT$(1)
260 IF C$="N" OR C$="n" THEN A$="NÃO":GOTO 80 ELSE IF C$="S" OR
C$="s" THEN A$="SIM":GOTO 270 ELSE 240
270 LOCATE0,7:PRINT"Digite o cabeçalho (máx.:60 caracteres)"
280 LINE INPUT D$:IF LEN(D$)>60 THEN PRINT:PRINT"Cabeçalho com m
ais de 60 caracteres":PRINT"Tecle algo" ELSE IF D$="" THEN 270 E
LSE 80
290 T$=INPUT$(1):GOTO 240
300 CLS:PRINT"OPÇÃO PARA IMPRESSÃO DE RODAPÉ"
310 LOCATE0,5:PRINT"Confirma impressão de rodapé (S/N) ?";:C$=IN
PUT$(1)
320 IF C$="N" OR C$="n" THEN B$="NÃO":F1=0:L$="NÃO":GOTO 80 ELSE
 IF C$="S" OR C$="s" THEN B$="SIM":GOTO 330 ELSE 300
330 LOCATE0,7:PRINT"Digite o rodapé (máx.:55 caracteres)"
340 LINE INPUT E$:IF LEN(E$)>55 THEN PRINT:PRINT"Rodapé com mais
 de 55 caracteres":PRINT"Tecle algo"ELSE IF E$=""THEN 300 ELSE 360
350 T$=INPUT$(1):GOTO 300
360 LOCATE0,11:PRINT"Inclui numeração (S/N) ?";
370 C$=INPUT$(1):IF C$="N" OR C$="n" THEN F1=0:F2=0:L$="NÃO":GOT
O 80 ELSE IF C$="S" OR C$="s" THEN F1=1:F2=0:L$="SIM":GOTO 80 EL
SE 360
380 CLS:PRINT"OPÇÃO PARA NUMERAÇÃO DAS PÁGINAS"
390 LOCATE0,5:PRINT"Confirma nova paginação (S/N) ?";
400 C$=INPUT$(1):IF C$="N" OR C$="n" THEN 80 ELSE IF C$="S" OR C
```

Figura 1 – Programa para imprimir programas fonte de maneira formatada.

```
$="s" THEN 410 ELSE 390
410 LOCATE0,7:PRINT"Digite 0 para não numerar páginas."
420 LOCATE0,9:INPUT"Qual o novo número da página ";A
430 IF A=0 THEN F2=0:L$="NÃO":F5=0:GOTO 80 ELSE IF A>0 THEN F2=1
:L$="SIM":R=A:F5=1:GOTO 80 ELSE 420
440 CLS:PRINT"OPÇÃO PARA Nº DE CARACTERES/LINHA"
450 LOCATE0,5:PRINT"Confirma redefinição (S/N) ?";
460 C$=INPUT$(1):IF C$="N" OR C$="n" THEN 80 ELSE IF C$="S" OR C
$="s" THEN 470 ELSE 450
470 LOCATE0,7:INPUT"Qual o novo nº de caracteres/linha ";B
480 IF B<0 THEN 470 ELSE 80
490 CLS:PRINT"OPÇÃO PARA Nº DE LINHAS/PÁGINA":LOCATE0,2:PRINT"Má
ximo aceito pelo programa: 55"
500 LOCATE0,5:PRINT"Confirma novo nº de linhas/pág. (S/N) ?";
510 C$=INPUT$(1):IF C$="N" OR C$="n" THEN 80 ELSE IF C$="S" OR C
$="s" THEN 520 ELSE 490
520 LOCATE0,7:INPUT"Qual o nº de linhas/página ";C
530 IF C<0 OR C>55 THEN LOCATE0,11:PRINT"Número de linhas/página
 ilegal":GOTO 520 ELSE 80
540 CLS:PRINT"OPÇÃO DE IMPRESSÃO DE PROGRAMAS (ASCII)"
550 LOCATE0,2:PRINT"Indique o drive (A/B) ";:F$=INPUT$(1):PRINT
F$:PRINT:PRINT
560 IF F$="A" OR F$="a" THEN FILES"A:" ELSE IF F$="B" OR F$="b"
THEN FILES "B:" ELSE 550
570 PRINT:PRINT:PRINT"Nome do arquivo:(<RETURN> sai da opção)"
580 LINE INPUT G$:IF G$="" THEN 80 ELSE G$=F$+CHR$(58)+G$
590 OPEN G$ FOR INPUT AS #1
600 CO=1:F4=0:A=1:PRINT:PRINT"Começa impressão na página (1): ";
:
610 INPUT CO:IF CO=0 THEN CO=1:A=1 ELSE IF CO=2 THEN A=2:CO=1:F4
=1 ELSE IF CO>2 THEN A=CO:CO=CO-1 ELSE IF CO<0 THEN PRINT:PRINT"
Valor ilegal":GOTO 600
620 TE=20:PRINT:PRINT"Termina na página (ultima): ";:
630 INPUT TE:IF TE<CO THEN TE=CO+1 ELSE TE=TE+1
640 IF F5=1 THEN PRINT:PRINT"Numeração redefinida para:";R:PRINT
"Confirma (S/N) ?"; ELSE 680
650 C$=INPUT$(1):IF C$="S" OR C$="s" THEN A=R:GOTO 690 ELSE IF C
$="N" OR C$="n" THEN 660 ELSE 640
660 PRINT:PRINT"Abandona impressão (S/N) ?;"
670 C$=INPUT$(1):IF C$="S" OR C$="s" THEN 80 ELSE IF C$="N" OR C
$="n" THEN 690 ELSE 660
680 PRINT:PRINT"Pausa entre as páginas (S/N) ?";
690 C$=INPUT$(1):IF C$="S" OR C$="s" THEN F6=1 ELSE IF C$="N" OR
 C$="n" THEN F6=0 ELSE 680
700 PRINT:PRINT:PRINT"Prepare a impressora e tecle algo";:T$=INP
UT$(1)
710 CLS:PRINT"[ESC] = sai da impressão":PRINT:PRINT"[STOP] = pau
sa/continua"
720 IF CO>1 OR F4=1 THEN GOSUB 950
730 IF A$="SIM" THEN LPRINT D$ ELSE LPRINT
740 LPRINT
750 X$=INKEY$:IF X$=CHR$(27) THEN CLOSE:GOTO 80
760 FOR I=1 TO C
770 IF EOF(1) THEN CLOSE:F3=1:GOTO 830
780 X$=INKEY$:IF X$=CHR$(27) THEN CLOSE:GOTO 80
```

```
790 LINE INPUT#1,H$
800 IF LEN(H$)>B THEN I=I+INT(LEN(H$)/B)
810 LPRINT H$
820 NEXT I
830 K=I-1
840 FOR L=1 TO 55-K:LPRINT:NEXT L
850 IF B$="SIM" THEN LPRINT TAB(B);E$;"    ";
860 IF F1=1 THEN LPRINT"Pág.:";A:GOTO 880
870 IF F2=1 THEN LPRINT TAB(39);A
880 A=A+1
890 IF F3=1 THEN F3=0:LPRINT CHR$(12):GOTO 80
900 LPRINT CHR$(12)
910 IF F6=1 THEN LOCATE0,5:PRINT"Tecle algo para continuar";:T$=
INPUT$(1):GOTO 730
920 IF A=TE THEN CLOSE:GOTO 80 ELSE 730
930 CLS:PRINT"Confirma saída do programa (S/N) ?";
940 C$=INPUT$(1):IF C$="N" OR C$="n" THEN 80 ELSE IF C$="S" OR C
$="s" THEN END ELSE 930
950 FOR O=1 TO C*CO
960 IF EOF(1) THEN CLOSE:GOTO 80
970 LINE INPUT#1,H$
980 NEXT O
990 RETURN
```

DICAS JOGO

Possuo um MSX 2.0 translormado e estou mandando as seguinles dicas:

– Hyper Sports 3

```
10 BLOAD "SPORT3-1": POKE &H94BE,0: DEFUSR=&D000: A=USR(0)
20 BLOAD "SPORT3-2", R
```

Obs.: Somente deixe apertado a seta para direita.

Golvellius – Megarom 1.0

Senha: ELFESOMTIF
M48J5GJ2T7

Obs.: com esla dica você já começa com

- Power ao máximo
- Find ao máximo
- Todos os cristais e objetos
- Faltando, apenas, matar Golvellius, cuja a localização é: esquerda, para cima, esquerda, entre pelo canal do rio para cima, destrua lodas as pedras em volta da tela e, quando lerminar, dê um golpe na pedra do meio, que abrirá uma passagem secrela. Passe por ela através do labirinto intermediário, para chegar ao último demônio. Para uma versão do Golvellius com deleito na última fase rode esle programa:

```
10 C$="GOLVELI": N=15: BLOAD "MEGARAM.BIN": DEEUSR=&HD100: A=USR(0): IF A THEM STOP ELSE FOR A=0 TO N: POKE &HDADE,A*2: A$=C$+HEX$(A)+".BIN": BLOAD A$: PRINT A$
15 IF A=0 THEN POKE &HAA80, 0· REM IMUNE!
20 IF A=1 THEM POKE &H9361,2: REM DEFEITO SONORO
30 DEFUSR = &HD1D3· X=USR(0) NEXT
35 FOR Z=0 TO 1400: NEXT Z
30 DEFUSR= &HD1FC: A=USR (0)
```

Super Laydock – Mission Stricker

Tire as linhas 15 e 20 do programa acima

Troque C$="GOLVELI" por C$="SLAYDOC"

Coloque:

```
15 IF A=1 THEN POKE &H9240 TO &H9242: POKE W,0: NEXT W
```

O restante do programa continua idêntico.

Obs.: Seu power não diminui, somente 1 nave.

Todas as armas: Password: 21121212121212

Clovis Palmira
Rua Maria Luiza Ferreira 24
06210 – Osasco – SP

Vocês publicaram o macete do Batman e El Mundo Perdido com imunidade total. Para que esses dois macetes funcionassem no meu micro, tive que fazer o seguinte:

El Mundo Perdido

```
10 SCREEN 2: POKE –1,177: COLOR 1,1,1: BLOAD "MUPERD-1",R: BLOAD "MUPERD-2": POKE &H998D,0: POKE &H998A,0: POKE &HB919,0: POKE &HB946,0: DEFUSR = &H9182: A=USR(0): BLOAD "MUPERD-3",R: BLOAD "MUPERD-4": BLOAD "MUPERD-5",R
```

No meu caso, o quarto bloco não pode ser executado, como loi publicado na revista.

Batman

```
1 'REVISTA CPU
2 'JOGO BATMAN
10 COLOR 15,1,1: SCREEN 2
20 BLOAD "BAT1",R
30 BLOAD "BAT2": POKE &HB36F,79: POKE &HB370,32: POKE &HC1C4,0: POKE &HB32E,77: POKE &HB36E,73: POKE &HB32F,65: POKE &HA999,0: POKE &HB330,82
40 DEFUSR=&HE00C: A=USR(0)
50 BLOAD "BAT3": POKE &H88EA,0
60 A=USR(0)
```

Inverti os Defusr, ou seja, no segundo bloco do Batman, o Defusr que executa é E00C e o terceiro bloco é executado pelo Delusr E00F.

E, para contribuir para a revista, quero dar o macete do jogo Ice World. Para jogar qualquer fase, digite CASIO, na opção Password.

Giles Leal de Lima
Rua Vicentina Goulart, 105 casa 3
Jardim Normandia
27250 – Volta Redonda – RJ

Estou mandando alguns macetes e gostaria que publicassem a resolução do jogo Gruta de Maquiné e do Avenger.

Valkyr

Ao carregar o jogo, pressione, simultaneamente, as teclas ESC, TAB, CONTROL, SHIFT, Seta para a direita, Seta para a esquerda e Seta para cima. Deverá aparecer na lela a palavra "cheat". Solle tudo e comece o jogo.

Knight Leon

Pressione, ao mesmo tempo, ESC, TAB, CONTROL, SHIFT, Y e barra de espaço. Para vidas inlinilas, aperte CAPS e barra de espaço.

TZR – Senhas
Pista 5 – Dusk
Pista 6 – Dark
Pista 7 – Blin
Pista 8 – Dawn

Kings Valley Plus – última fase

Pressione Control, K e digite YOSHITATAKA ITOH

Fábio Lins Leite
Rua Barão de Serro Largo, 33
Irajá
21237 – Rio de Janeiro – RJ

Estou enviando dicas de jogos, para que sejam publicadas na seção de cartas:

Zanac 2 – Super dica

98 vidas – Seta para a direita, esquerda, baixo e y.

Para ter qualquer arma, basta pressionar o número correspondete, de 1 a 7.

Ease II com 98 vidas – Shilt, Conlrol, TAB, ESC, HOME/CLS, seta para a esquerda, direita e baixo. Assim que você entrar no round Il, aparecerão 6 Totens enlileirados. Cada um que você acertar solta uma bola amarela. Esta bola, com o lempo, escurece e, quando você tocar na bola preta, ela lhe levará para outras fases, na seguinte ordem:

1º – Round 8
2º – Round 12 – o melhor
3º – Round 7
4º – Round 4
5º – Round 2
6º – Round 9

Robocop – Para passar de fase
A qualquer hora do jogo, pressione as teclas: ESC, TAB, CONTROL, SHIFT, D, F
Esta dica só não dá certo na fase que monta as caras. No resto do jogo funciona.

Galt Force – 11 poderes e 9 vidas
Escolha o personagem primeiro e aperte as teclas 1, 5, 0 e espaço.

Hunch Back – Vida Infinita
```
10 BLOAD "NOME": POKE
-28370,0
20 DEFUSR = &H9000: A = USR(0)
20 RUN
```

Back to the future – Vida Infinita
```
10 BLOAD "NOME": POKE
&H908B,255: POKE &H90C9,255
20 DEFUSR=&HD000: A=USR(0)
30 BLOAD "NOME",R
```

Zanac 1 – Vida Infinita
```
10 BLOAD "NOME": POKE
&H9654,0: DEFUSR = &HD000:
U=USR(0)
20 BLOAD "NOME",R
```

Scion – 80 Vidas
Na tela de apresentação, basta apertar 1,5 e 0 (ao mesmo tempo)

Gutt Blaster – Vida Infinita
Tecle STOP e digite COSMIC. Na tela, aparecerá a frase "Programmers Mode On", que significa que o jogo está com vida infinita.

Giovanni G. Junior
Rua Martinho Prado 211 apto 142
01306 – São Paulo – SP

Venho comprando a revista CPU desde o número 6, quando soube de sua existência e gostei muito.

Quando tentei concluir o jogo "O Conde de Monte Cristo", pelo roteiro da página 58, o personagem não pegava a pá, pois não havia mais lugar. A solução é não pegar a colher, que não tem nenhuma utilidade.

Onde está a instrução "Destranque Arca", substitua por "Use grampo".

Sérgio de Oliveira Delocco
Rua Rio da Prata 1647
Bangu
21830 – Rio de Janeiro – RJ

Tenho apreciado muito a revista que editam, que tem fornecido farto material de esclarecimento em diversas questões. Contudo, como tenho algumas dúvidas, indago se é possível a divulgação de "O segredo do Atlântida" e dicas de jogos, entre os quais "Fuzzball", "Halloween", "Spelunker", etc.

Sugiro, ainda, que promovam concursos voltados para as diversas faixas etárias de seus leitores. Eu, por exemplo, tenho 11 anos e estou na sexta série. Ofereçam prêmios, como megaram, drive, jogos, etc.

Aproveito para deixar a minha colaboração, representada pela dica dos seguintes jogos:

Starquake – Códigos de teletransporte: Razon, Taraq, Vorax, Dulan, Antio, Angor, Kwake, Ercol, Uplan, Kranz, Indle, Snoot, Optin, Zodia, Argol.

Kings Valley Plus – Fase 16: Crystal ball – Fase 31: Nile river – Fase 46: King ship – Fase 60: Yoshitaka itoh. Aperte Control e K para acessar as senhas.

Turmoil e Prédio Assombrado – Aperte Select para mudar de fase.

André Gustavo Caixete Marinho
Caixa Postal 6231
70740 – Brasília – DF

Nós somos leitores de CPU desde o número 3 e sentimos que neste espaço de tempo a revista evoluiu bastante, não só na qualidade de seus artigos como na parte gráfica, apesar de poder melhorar ainda mais. De modo geral, ela está ótima.

Podemos sentir que nesta revista há um grande espaço e muita liberdade para os leitores exporem suas idéias e isso é um ponto muito positivo, pois assim pode-se ouvir a voz de todos. Resolvemos mandar nossa contribuição.

Se possível, gostaríamos que dessem uma explicação melhor sobre os comandos Play, Sound e que falassem, também, sobre Sprites.

Corsários
Após ter dado início ao jogo, pressione, ao mesmo tempo, as teclas F, V e U. Aparentemente, nada acontecerá, mas ao tomar a primeira pancada, ao invés de diminuir, sua energia subirá para 999. Só a partir da segunda pancada sua energia começará a diminuir. Isto poderá ser repetido a qualquer instante do jogo.

Mutant Zone 1 e 2
Após iniciar o jogo, pressione, ao mesmo tempo, as teclas S, D, L, M ; R?. Este macete lhe confere imunidade.

The Legions Xevious
```
10 REM THE LEGION XEVIOUS –
VIDAS INFINITAS
20 BLOAD "XEVIO1": POKE
&H92D7,0
30 DEFUSR = &HD000: A=USR(0)
40 BLOAD "XEVIOI2",R
```

Marco Aurélio G. Araujo
Fernando P. Figueiredo
Estrada Engenho da Pedra, 1333/201
Olaria
Rio de Janeiro - RJ

Estou enviando a dica do jogo Gyrodine.

Gyrodine – Escolha de fase para início de jogo
Na tela de abertura, mantenha pressionado ESC, SELECT e depois tecle a barra de espaço. Não solte ESC+SELECT antes de aparecer a tela de opções. Escolha sua tela pressionando as teclas dos cursores cima-baixo.

Rogério Bello dos Santos
Rua Chady Muradi 81
Jaguare
05351 – São Paulo – SP

Como leitor da revista CPU, gostaria que publicassem as dicas e mapas para os seguintes jogos: Hércules, Goody, Feud, Camelot, Demonia, Starquake e Paris Dakar.

Yie ar Kung fu 2
Para conseguir 94 vidas, aperte a

tecla "E" 1 vez, "S" 2 vezes, "C" 3 vezes, "F" 4 vezes, ou seja, ESSCCCFFFF. Mas faça isso rápido, antes da apresentação começar.

Titanic I
Pressione, simultaneamente, 2, 3, 5 e 8, após o início do jogo, para obter vidas infinitas.

Titanic II
O código é Susie e, para obter vidas infinitas, proceda do mesmo modo que no Titanic I.

Sol Negro 2
O código 2414520

Frederico B. Lage
Av. Lineu de Paula Machado, 1006/1302
Jardim Botânico
Rio de Janeiro – RJ

TROCA DE CORRESPONDÊNCIA

Estou interessado na obtenção de material de qualquer tipo sobre linguagens de programação, como Turbo Pascal, C e Cobol.

Gostaria que vocês me informassem sobre profissionais que obtiveram algum sucesso com estas linguagens no MSX e seus respectivos trabalhos e sugiro que sejam feitas mais reportagens sobre o assunto.

Marcos Job Anghinoni
Av. Dom Pedro II nº 1999 apto 14
Bairro Campestre
09000 – Santo André – SP

Atualmente, existem vários programas para MSX desenvolvidos em Pascal e outras linguagens sem ser o Basic. Como exemplo, podemos citar o programa Top Cad.

Sempre que possível, publicamos em CPU programas em outras linguagens.

Gostaria de trocar programas, dicas e informações com outros usuários de MSX 1.0 e 2.0, com ou sem Megaram, que possuam drive de 5 1/4.

José Maurício Machado
R. Prof. Antônio Nascimento, 67
09820 – São Bernardo do Campo – SP

Gostaria de trocar jogos e manuais com usuários de MSX com drive.

Nadia de Souza
Rua Braz de Francesco 100 B2 apto 402
60325 – Fortaleza – CE

Gostaria de saber o código para que eu possa jogar o "Capitão Sevilha 2" e poderiam publicar dicas para os jogos Tom e Jerry, Freddy Hardest 2, Comand 4 e Casanova.

Delcio da Costa Peçanha Junior
Rua Gregório de Castro Moraes, 453/101
Ilha do Governador
21931 – Rio de Janeiro – RJ

Gostaria de corresponder-me com outros usuários da linha MSX, para troca de programas e dicas para MSX 2.0, com ou sem Megaram.

Marcelo Rodrigues
R. Aureliano Coutinho 108/31
São Paulo – SP

Gostaria de trocar correspondência com leitores interessados em dicas de jogos ou programas para MSX.

Possuo um Hotbit versão 1.1 e um Data-Corder da Gradiente.

Hélio Yamamoto Fuck
R. Presidente Lima 620
Centro
29100 – Vila Velha – ES

MSX?

Porque o nome MSX? O que significa este nome? Quando foi lançada a linha MSX?

Alexandre de O. Fernandes
Rua Vistula 106
21931 – Guarabu – RJ

O nome MSX significa Microsoft Extended. MS da Microsoft e o X de Extended. A Microsoft, em conjunto com outras empresas japonesas, resolveu criar um padrão de microcomputadores. Hoje em dia, este padrão já está na terceira versão, a 3.0, que está prestes a ser lançada lá fora. O primeiro micro da linha MSX foi lançado em 1985, no Brasil.

MSX 2.0 RGB

Estou prestes a transformar o meu MSX 1 em 2.0 e gostaria de saber se eu comprar uma TV a cores com entrada RGB embutida poderei utilizá-la com o meu MSX 2.0 e também para assistir os programas normais que são transmitidos pela televisão. E caso transforme minha TV para que tenha saída RGB também poderei fazer o mesmo?

No jogo Alcatraz existe uma parte do jogo que vocês escreveram "Puxe corda" mas, antes de puxar a corda, nós devemos comandar "Jogue Corda", para que funcione.

Aproveito para enviar algumas dicas de jogos:

Safari X – Vidas Infinitas
BLOAD "CAS:":POKE &H9065,0: DEFUSR = &H8EE0: A=USR(0)

Fire Star – Vidas Infinitas
Incremente este POKE no último bloco, depois da instrução NEXT – POKE &H8896,0

Eric and the Floters – Vidas Infinitas
BLOAD "CAS:":POKE &HC0066,0: DEFUSR =&HE040: A=USR(0)

Giles Leal de Lima
Rua Vicente Goulart 105 casa 3
Jardim Normandia
27250 – Volda Redonda – RJ

Para verificar se a televisão com saída RGB que pretende comprar pode ser utilizada com o seu MSX 2.0, verifique junto aos fabricantes, tanto do televisor quanto da placa, as características técnicas, para verificar se são compatíveis, o mesmo acontecendo caso faça a adaptação em sua televisão antiga. Você poderá utilizar a sua televisão normalmente, mesmo tendo o computador ligado à saída de RGB.

PIRATARIA

Não se sabe quando e como essa doença, a pirataria, vai acabar.

Pirataria é a obtenção de um programa sem pagar os devidos direitos autorais para a pessoa que o fez, seja em disco ou em fita.

No Brasil, o problema de pirataria é muito acentuado. A cópia de jogos não é considerado pirataria neste país, pois os jogos não são registrados na SEI, salvo excessões.

Os piratas alegam que praticam a pirataria porque os preços são caros e é justamente por causa da pirataria que os custos de software nacional são elevados.

O autor sabe que venderá poucas unidades de seu programa, por causa da pirataria e, para poder compensar o tempo que dedicou ao programa, tem que elevar o preço. Por outro lado, os usuários, ao invés de comprar o software legalmente, pagando direitos autorais, recebendo o programa com suporte, manual, etc., prefere ir a casa de um amigo e copiar, para desespero do autor.

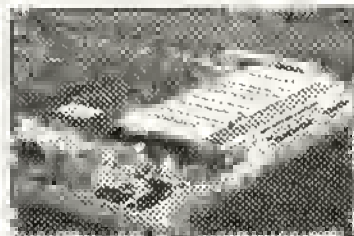
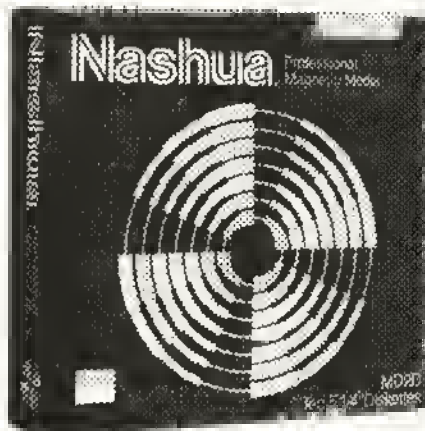
Uma outra forma de o autor se proteger é vender as cópias travadas, numa tentativa de evitar a cópia não autorizada. Trava que lhe consome mais tempo e acaba elevando mais o custo do programa.

Se a pirataria acabasse, os usuários passariam a comprar os programas originais, que custariam bem mais barato, pois o autor venderia bem mais.

Várias pirato-houses, que geralmente tem pequeno porte, vendem softs registrados, originais, a preços simplesmente ridículos, que muitas vezes não passam de 10% do preço do software. E esses 10% eles enfiam no bolso.

Na realidade, o pirata é um ladrão.

Sugiro que as grandes softhouses ofereçam prêmios a quem denunciar os piratas para que possamos diminuir, já que acabar seria utopia minha, que só atrapalham o mercado e não ajudam ninguém, muito menos aqueles que compram software em suas mãos, pois ficam totalmente sem suporte.

Célio Wakamatsu
Rua Albuquerque Lins 772/101
01230 – São Paulo – SP

BIBLIOTECAS EM TURBO PASCAL
Parte II

Frederico dos Santos Liporace

Nesta segunda parte, trataremos dos procedimentos de acesso ao interpretador BASIC residente na ROM. Uma característica comum das rotinas até aqui conhecidas, que permitiam esse acesso, era a de que, se houvesse algum erro durante a interpretação do comando BASIC desejado, seu programa ia para o espaço, ou, eufemisticamente, "o controle ficava com o interpretador".

Uma solução para esse problema seria fazer uma análise prévia do comando BASIC antes de mandá-lo para o interpretador. Isso, no entanto, implicava numa piora tanto na quantidade de memória utilizada quanto na velocidade de processamento, além de ficar particularmente dificultada em comandos com sintaxe complexa como DRAW e PLAY. Também não podíamos usar,com segurança,instruções do tipo FILES"*.PIC" quando não houvesse nenhum arquivo no disco com a extensão PIC, etc.

O procedimento proposto nesse 2º artigo da série possui as seguintes características:

– Não existe a necessidade de se definir um procedimento para cada instrução do BASIC.

– Retorna ao Pascal seja qual for o tipo de erro ocorrido durante a interpretação, inclusive o pressionamento de CTRL+STOP.

É importante lembrar que o procedimento CLPRIM, publicado no artigo anterior, deve ser incluído durante a compilação. O funcionamento da rotina é bastante simples:

– O argumento do procedimento Basic é uma string de no máximo 255 caracteres, contendo os comandos que o BASIC irá executar.

– É adicionado o código 0 à linha, visando informar ao interpretador onde esta termina.

– Coloca-se no "par de registradores" HL o endereço do início da linha, através de INLINE (poderíamos obter efeito equivalente usando a função Addr).

– Através de CLPRIM, transfere-se o controle para o Loop Principal (Mainloop) do interpretador, no endereço imediatamente após a chamada da rotina da BIOS que o interpretador usa para receber uma linha do teclado, quando no modo direto do BASIC. Essa rotina, chamada PINLIN, situa-se no endereço 00AEH e fornece como saída o endereço da linha recolhida no par de registradores HL.

Já deu para perceber o truque, não?

Criamos uma linha "falsa", colocamos em HL seu endereço inicial e transferimos o controle para o interpretador BASIC. Este "pensa" que a linha foi obtida por PINLIN e a executa como se estivéssemos no modo direto. Falta explicar como é feito o retorno ao Pascal em caso de erro de interpretação ou pressionamento de CTRL+STOP.

Uma característica peculiar de CLPRIM é a de permitir o retorno à ela em qualquer momento, seja qual for a posição do STACK, através de uma rotina específica. A variável HANDLER contém essa rotina de retorno, e basta instalar, através do procedimento InstHook, nos ganchos correspondentes ao final de linha (FEF8H), ocorrência de erro de sintaxe (FFB1H) e pressionamento de CTRL+STOP (FFO2H) um jump para o endereço dessa variável (obtido através de Addr).

Note que como não foi usado nenhuma vez ABSOLUTE, consegue-se a flexibilidade ideal à rotina.

Com isso, integramos o BASIC sem traumas ao ambiente do Turbo-Pascal. Mês que vem, mostraremos que é possível fazer o mesmo com o Assembler.

Até lá!

```
PROGRAMA EXEMPLO
{$i clprim.inc}  { Inclui CLPRIM }
{$i basic.inc}   { Inclui BASIC }

begin
  InstalBasic; { Instala os hooks e ini- }
               { cializa Handler }
  Basic ('PLAY"L4GGO5L2CO4L4BAL2GL8FL4FL8AL4AAGF'+
         'L2EL8DL4DL8FL4FFL8EL4DL8CL4C.L8DCL4DL8'+
         'CCL4C.","L1O3GBAGAA-L4A.L8GL2G","L1O3C'+
         'GFCFFL4F.L8EL2E":?"Lennon/McCartney"');
  { A linha foi dividida apenas para facilitar }
  { a leitura. Experimente CTRL+STOP }
end.

BASIC.INC
type
  LinhaBas=String[255];
  { Esta variavel vai conter a }
  { linha de BASIC }

var
  { Essa variavel vai conter a }
  { rotina de retorno a CLPRIM }
  Handler:String [7];

Procedure InstHook (x:integer);
  { x e o endereco do gancho }
  begin
    Mem[x]:=$C3; { Jump... }
    { Endereco de Handler }
    Mem[x+1]:=Lo(Addr(Handler));
    Mem[x+2]:=Hi(Addr(Handler))
  end;

Procedure Basic (x:LinhaBas);

  begin
    x:=x+#0; { Indica fim de linha }
    InLine ($21/x/$22/HL);
    { HL = Inicio da linha }
    Clprim ($4167) { BASIC!... }
  end;

Procedure InstalBasic;
  begin
    Handler:=#$ED+#$7B+#$9A+#$F3+
             #$C3+#$92+#$F3;
    { Rotina de retorno para Clrprim }
    { Instala ganchos para ... }
    InstHook ($FF02); { CTRL+STOP }
    InstHook ($FFB1); { Erro Interpretador }
    InstHook ($FEF8)  { Fim de programa }
  end;
```

PROJETO MSXDEBUG

PARTE IX

Sérgio Duarte Calheiros

A implementação do comando APPEND, feita na última parte do projeto, funcionou como ponto inicial de uma fase, ou melhor, de um sub-projeto, vinculado ao projeto do MSXDEBUG.

A partir deste ponto, o MSXDEBUG estará sendo preparado para auxiliar o leitor a lidar com um campo do computador ainda pouco explorado, de forma relativamente facilitada. A re-integração de programas divididos, tema introduzido na parte anterior, passará a ser o novo problema dos usuários do MSX.

Neste mês, como antecipamos, implementaremos o comando SAVECOM. Ao contrário do procedimento normal, desta vez começaremos logo com a implementação do comando ao MSXDEBUG. Isto é necessário porque, neste caso, a exemplificação é mais importante que apenas a descrição.

O código destinado ao comando, listado no bloco 1, tem local fixo, sendo necessário respeitar o endereço dado. Além deste pequeno detalhe, vale apenas recomendar a atenção que sempre deve ser dispensada na digitação de um bloco de códigos. Na implementação do comando na tabela, o BYTE indicador do fim da tabela de comandos deve ser deslocado para o endereço 4D25H, assumindo que o leitor tenha carregado todo o programa a partir do endereço 4100H. Nos endereços 4C9EH e 4C9FH digite o endereço de entrada, que no caso é 17EEH, ou seja, BYTE 0EEH em 4C9EH e BYTE 17H em 4C9FH.

Terminada a digitação, verifique a SOMA do bloco, atualize a versão do MSXDEBUG para 1.7 e salve. Com o comando em mãos, será mais fácil acompanhar seu funcionamento e uso.

O comando SAVECOM será o responsável pela gravação de programas da memória para o disco, preparando-os para rodarem sob o ambiente do DOS. Além de gravar o programa no disco, atribuindo a extensão COM ao arquivo criado, o comando adiciona uma pequena rotina ao programa. Esta rotina, mais comumente chamada RUNTIME, permanece responsável em assegurar a correta execução do programa COM. Mais especificamente, o que o RUNTIME faz, é esperar o drive desligar, transferir o programa para o seu respectivo local, preparar o ambiente que o programa deverá encontrar ao ser carregado e, finalmente, executá-lo.

O processo de reintegração dos programas, ou conversão, para simplificar o linguajar, envolve várias etapas, entre as quais o uso do comando SAVECOM. Seu uso constitui a última etapa do processo, sendo responsável apenas pela gravação do programa inteiro, já reintegrado e unido pela memória.

As demais fases do processo dependerão somente do usuário, ou melhor, da prática e do conhecimento de cada um. É fundamental que os leitores possuam uma visão um tanto clínica do problema, já que não há uma receita única capaz de descrever exatamente cada passo a ser tomado pelo operador. Entretanto, existe um roteiro geral para a conversão, que deverá ser sempre seguido pelo leitor. Raciocinando logicamente, tentemos acompanhar o mecanismo que está por trás do processo de carregamento, união e execução dos programas que estão separados em vários blocos.

É sabido que, no MSX, é possível chavear páginas de memória num mesmo local. Sabemos, inclusive, que o BASIC, responsável pelo carregamento da maioria dos programas a serem convertidos, tem acesso a somente uma parte reduzida desta memória. Ora, somos forçados a concluir que o papel do BASIC se resume apenas em carregar os blocos e passar o controle para a rotina responsável em transferir a parte do programa em questão para seu devido local. Os blocos são carregados seqüencialmente na mesma área e transferidos para seus respectivos lugares, sejam quais forem, até que o programa esteja completo para ser executado, o que acontece no último bloco. Este é o ponto crucial do problema. Como evitar a execução, logo após o programa estar carregado em seu lugar? A resposta é simples: agir como o BASIC, simulando, ou melhor, imitando seus passos, até que atinjamos o ponto desejado, antes da execução. Desta maneira, podemos decidir se desejamos executá-lo ou, então, salvá-lo em disco, com o comando SAVECOM.

As informações vitais, como as partes úteis de cada bloco em si, ou seja, os BYTES que pertencem ao programa e não à rotina de transferência, o local real destes blocos e o ponto de execução de todo programa, quando completo, serão encontradas nas rotinas de transferência.

O endereço de execução final, só é encontrado no carregamento do último bloco. Os demais blocos sempre devolvem o controle ao BASIC ao fim de suas tarefas de habilitação da memória e da transferência. Não importa como estes blocos são transferidos, o que importa é que as informações estão lá, prontas para serem lidas.

A maior facilidade do processo, através do MSXDEBUG, está no fato do programador dispor de uma memória linear e também acesso ás áreas não alcançadas pelo BASIC. A maior dificuldade, muitas vezes superada, está no tamanho da memória. Num lado está o MSXDEBUG, ocupando a parte mais baixa da RAM. De outro está o DOS, ocupando a parte mais alta. Quando não houver espaço suficiente, deveremos lançar mão de outros recursos, inclusive do comando APPEND. O processo de conversão de programas longos é perfeitamente viável, mas dependerá unicamente da prática do usuário. Felizmente, a maioria dos programas tem um tamanho compatível com o espaço disponível.

Vejamos, então, como proceder em linhas gerais. O primeiro passo é determinar se o programa tem algum

macete no carregamento. Normalmente, basta um comando como RUN "programa" para executar um programa com vários blocos ou digitar BLOAD "programa", R, caso o programa esteja num único bloco.

No caso de ser um único bloco, (preste atenção) as informações referentes ao local real do programa estarão na rotina indicada pelo endereço de entrada, localizado no header do arquivo binário gravado pelo BASIC. Caso sejam vários blocos, devemos nos certificar se todos os blocos são carregados com execução automática (BLOAD "programa",R) ou se são apenas carregados (BLOAD "programa") e posteriormente executados (DEFUSR=XXXX:A=USR(0)), num ponto independente do endereço de entrada contido no header do arquivo binário criado pelo BASIC. Quando a execução for automática, devemos seguir o endereço contido no header, caso contrário, o definido na instrução DEFUSR. É importante assegurar que isto foi bem entendido!

O termo "seguir o endereço" se refere ao procedimento de rastrear a rotina responsável pela transferência do bloco, justamente para descobrir o local real daquela parte do programa.

Estando com o programa completo em seu lugar, resta salvá-lo inteiro, usando o comando SAVECOM, sabendo como preparar o ambiente do programa.

O grau de complexidade do processo envolvido na conversão de um determinado programa pode ser praticamente nulo ou mesmo tão complexo que nem um programador experiente consegue fazê-lo. Entretanto, ao invés de continuar a teorizar o assunto, entrando em mais detalhes, daremos exemplos práticos de como se constitui o processo da conversão.

Os programas mais comuns a serem convertidos que o usuário deverá encontrar, são os jogos e os aplicativos retirados de cartuchos. Estes programas tem uma característica particular, que é a posição de memória que ocupam. Além disso, são programas binários, que quase sempre dispensam o interpretador BASIC para funcionar. Aliás, a conversão de programas BASIC constitui mais um ramo desta árvore, que será discutido mais tarde.

Os programas mais simples de serem convertidos são aqueles que contém apenas 1 único bloco de até 22 Kbytes. Jogos antigos, como os da KONAMI, são os mais fáceis. Todo usuário de MSX já deve ter usado estes programas. Entre eles estão vários, como HYPERSPORTS 1 e 2, ROLLER BALL, DECATHLON, MACACO ACADÊMICO, etc. . . Dispor de algum destes programas já é mais que suficiente para iniciar a sua primeira conversão.

Programas deste tipo, geralmente, ocupam o mesmo local. Rodam em 4000H ou 8000H, páginas 1 e 2, respectivamente. Para descobrir o local correto, só explorando o conteúdo do programa.

Não esqueça de observar o detalhe de como o programa é carregado. Geralmente, basta entrar no BASIC e usar o comando BLOAD "programa",R. Automaticamente, o programa é carregado e executado. Se o programa que você escolheu para ser convertido obedece a este quesito, siga em frente, senão arranje outro.

Inicialmente, estando no MSXDEBUG, carregue o programa com o comando BLOAD e anote os endereços fornecidos. Estes endereços correspondem ao início, fim e ponto de execução do bloco. O mais importante deles é o último, que indica onde devemos seguir o código do programa. Neste ponto da memória, ou melhor, do programa, está a rotina que habilita a página de memória e transfere o bloco que interessa para seu respectivo lugar.

A seguir, use o comando DASS, para ler a rotina de transferência. Use, é claro, o terceiro endereço fornecido pelo comando BLOAD. Imediatamente, uma série de instruções devem surgir na tela. Não se preocupem em entender o que realmente significa cada uma, apenas procure pela instrução LDIR do assembler, cujo código binário é EDB0. Esta é uma macro instrução, ou seja, executa vários passos de uma só vez. Ela é a responsável pela transferência interna de um bloco de dados. Seu funcionamento é relativamente simples, agindo em conjunto com os registradores do Z-80, mais especificamente, os registradores BC, DE e HL. Em BC deve estar o número de BYTES que devem ser transferidos. Em DE deve estar o primeiro endereço para onde o bloco deve deslocar-se, isto é, o endereço de destino. Finalmente, em HL deve estar o endereço do primeiro BYTE do bloco em questão, isto é, o endereo de origem. Com estes três valores, já temos em mãos as informações do local original do programa, tamanho e destino. Observe que a ordem dos registradores não importa. Importa apenas que os registradores sejam inicializados antes do LDIR, de forma semelhante ao trecho abaixo:

```
LD BC, quantidade
LD DE, destino
LD HL, origem
LDIR
```

Para calcular os endereços efetivos, o procedimento é simples. O registrador HL já traz o endereço de origem e DE o endereço de destino. Para saber até que ponto o bloco se estende, basta somar o valor que está em BC com o valor que está em HL. Suponhamos que o endereço que indica o fim do bloco seja fim = origem + quantidade. Então, descobertos os endereços do bloco, resta movê-lo para seu respectivo local. Para isso, use o comando do MSXDEBUG: MOVE origem fim destino.

O próximo passo, como o programa possui apenas um bloco, é descobrir o endereço de execução do programa. Geralmente, existe uma instrução de desvio do Z-80 diretamente para o ponto desejado. Esta instrução é o JUMP, encontrada logo após o LDIR ou próximo a ele. O mnemônico é JP e o código é C3. Anote o endereço mostrado após o JP. Observe que o valor mostrado após o código C3 está invertido.

Neste momento, o programa já se encontra pronto para ser gravado. Entretanto, antes de usar o comando SAVECOM, ainda vale discutir mais um detalhe. Para simplificar, suponhamos que o programa que está sendo convertido ocupe a página 1, ou seja, vai de 4000H a 7FFFH, exatos 16 Kbytes. Como explicamos na parte anterior, esta quantidade de BYTES ocupará 16 clusters no disco. Quando o bloco for gravado com o comando SAVECOM, lhe será acrescentado a rotina de transferência com exatos 48 BYTES. Isto implicará na ocupação de mais um

cluster no disco. Para contornar este problema, existe ainda um recurso. Não existe regra que determina que o programa ocupa exatamente 16 Kbytes. Dependendo do programa, alguns BYTES ao fim da página não são usados. Quando isso acontece, podemos diminuir o bloco para seu tamanho real, sem ocupar espaço útil do disco com dados inúteis. Com isto, podemos dispor do espaço para a rotina de transferência, sem ocupar mais um cluster no disco.

Para saber até que ponto vai o programa, basta explorá-lo. Para isso, use o comando DISP. Ainda supondo que o programa ocupa a página 1, comece a explorar a partir da metade (endereço 6000H), sempre avançando. O que deve aparecer, inicialmente, são o códigos do programa. Quando chegar perto do final, fique atento ao surgimento de um padrão único, ou seja, a repetição de um valor único até o fim. Se isso ocorrer, é sinal que esta área não é usada pelo programa, podendo ser descartada. Se, por exemplo, o programa for de 4000H até, digamos, 7FB7H, é sinal que conseguimos descartar a quantidade de BYTE suficiente para liberar o cluster que seria ocupado no disco. Naturalmente, se mais espaço for liberado, tanto melhor. Muito cuidado para não descartar dados que estejam no fim do programa, pensando que é código inútil. Se você não se sentir seguro, procure fazer isso mais tarde.

Agora entra a fase que determina como o ambiente deve ser preparado para receber o programa. Isto é necessário, pois, como dito, cada programa requer um ambiente diferente. O termo ambiente traduz tudo aquilo que o programa espera encontrar para funcionar, como a disponibilidade de memória ROM ou RAM, posição do STACK, etc. . . Alguns programas exigem que o ambiente de trabalho seja o mais específico possível, enquanto outros permitem uma certa flexibilidade.

O uso do SAVECOM envolve muito mais conhecimento por parte do usuário do que pode parecer. Além de fornecer os parâmetros referentes à localização e execução, é necessário estabelecer os outros parâmetros que auxiliarão na preparação do ambiente para o programa.

Quando o programa é carregado na memória pelo DOS, toda a memória RAM, de 64 Kbytes está habilitada. O programa em questão é colocado a partir do endereço 100H e aí mesmo executado. O que o micro passará a executar, será, então, o nosso programa, no caso, o RUNTIME acrescentado pelo comando SAVECOM.

Relembrando, a primeira atitude tomada pelo RUNTIME é esperar pelo desligamento do drive, para depois transferir o programa para seu local original. Antes de executar o programa já transferido, é necessário agora habilitar as páginas de memória ROM, que o programa precisará para funcionar. Muitos programas, talvez a maioria deles, precisarão somente da BIOS, ou seja, da ROM da página 0 para funcionar. Alguns poderão precisar, além da BIOS, de rotinas do próprio interpretador BASIC, devendo então serem habilitados. Por último, existem aqueles que, por ocuparem a RAM da página 0, não precisam da BIOS nem do BASIC, não devendo fazer nenhuma mudança na configuração do momento.

Outro ponto a ser definido é a localização do STACK POINTER do Z-80. O stack pointer é o ponteiro de pilha, responsável pelo controle das subrotinas em linguagem de máquina. Em geral, o STACK POINTER ocupa uma das posições mais altas da RAM. Na maioria das vezes, não é necessário mudar a localização do STACK, pois, a maioria dos programas não tem tamanho suficiente para atingi-lo. Entretanto, caso o programa seja excessivamente grande, ou caso a memória do computador esteja muito carregada, pode ser necessário deslocá-lo, para que não haja colisão do programa com o STACK. Isto poderia destruir o código do programa, fazendo com que seu funcionamento se torne irregular ou mesmo não funcione.

No caso do programa que estamos convertendo, com apenas um bloco, não teremos que nos preocupar com o STACK agora. Como supomos que seja um programa de cartucho, acreditamos que o código não será jamais atingido.

Um terceiro parâmetro, é o OFFSET, que permite dizer ao comando se o programa que está sendo salvo em determinada posição da memória deve ser executado em outra.

Neste momento chega a hora de utilizarmos o comando propriamente dito. Desta maneira, os procedimentos junto ao comando SAVECOM se resumem em passar os parâmetros corretos. Primeiramente, devemos estabelecer os endereços inicial, final e de execução do programa em questão, como no comando BSAVE. Com o programa supostamente ocupando a página 1 (4000H a 7FFFH) e com endereço inicial, digamos, em 4010H, usamos o comando como mostra a seqüência de parâmetros abaixo:

```
SIS > SAVECOM programa 4000 7FFF 4010
STACK>
OFFSET>
AMBIENTE > FC
```

Logo após a digitação do comando com os parâmetros iniciais, surgirá a pergunta sobre o posicionamento do STACK POINTER. Teclando <CR> somente, ou seja, sem indicar a nova posição, o programa assumirá que não deve mudar o valor do STACK no momento da execução do RUNTIME. Deste modo, o STACK continuará com seu valor original. A seguir, o usuário deve informar o OFFSET. Neste momento, não é necessário utilizá-lo e, mais adiante, veremos como funciona e para que serve este parâmetro. Apenas tecle <CR> sem mais problemas. Finalmente, chega o momento de definir o ambiente, ou seja, que memórias estarão habilitadas e quais não. O parâmetro a ser passado é um valor numérico de 8 bits, mas apenas três deles serão usados, de acordo com a tabela abaixo:

FF – Configurar RAM nas páginas 0 e 1
FC – Configurar ROM na página 0 e RAM na página 1
F0 – Configurar ROM nas páginas 0 e 1

Baseado nos valores da tabela acima, podemos concluir que o valor a ser utilizado deve ser FC, já que desejamos habilitar somente a ROM da página 0, ou seja, a BIOS, deixando a RAM da página 1, que vai de 4000H a 7FFFH, para o próprio programa.

Ao fim desta seqüência, o comando salvará o programa no disco, pronto para ser executado, rapidamente e ainda livre do BASIC. Execute-o e verifique se o processo foi bem sucedido.

Devido à complexidade do comando, aconselhamos que este artigo seja lido

tantas vezes quanto necessário. No próximo número, entraremos em aspectos técnicos do comando e, também, na explicação do parâmetro OFFSET.

Até lá.

```
Bloco 1

57EE CD A5 Ø8 CD FA Ø8 22 Ø5
57F6 19 11 AF 18 CD 18 ØB CD
57FE 24 ØB CD 62 ØA 21 ØØ ØØ
58Ø6 E5 CD 27 Ø9 28 Ø4 CD FA
58ØE Ø8 E3 E1 22 D4 18 7C B5
5816 3E Ø1 28 Ø2 3E 31 32 D3
581E 18 11 B5 18 CD 18 ØB CD
5826 24 ØB CD 62 ØA 21 ØØ ØØ
582E E5 CD 27 Ø9 28 Ø4 CD FA
5836 Ø8 E3 C1 2A 89 ØD Ø9 22
583E EØ 18 2A 8B ØD Ø9 22 DB
5846 18 11 BC 18 CD 18 ØB CD
584E 24 ØB CD 62 ØA CD 9A Ø8
5856 CD FA Ø8 7D 32 Ø1 19 3E
585E 2A 32 D6 18 21 Ø1 ØØ 22
5866 D7 18 21 48 Ø1 22 E7 18
586E 21 ØØ Ø8 22 C7 18 CD 9F
5876 18 CD 66 Ø7 CD 72 Ø7 CD
587E 5D Ø6 11 8Ø ØØ 21 C5 18
5886 Ø1 48 ØØ ED BØ 2A 89 ØD
588E Ø1 38 ØØ ED BØ 22 89 ØD
5896 11 5C ØØ CD 2B Ø7 C3 89
589E Ø6 11 AC 18 21 1F ØE EB
58A6 Ø1 Ø3 ØØ ED BØ C9 43 4F
58AE 4D 53 54 41 43 4B ØØ 4F
58B6 46 46 53 45 54 ØØ 41 4D
58BE 42 49 45 4E 54 45 ØØ FB
58C6 11 ØØ ØØ Ø6 ØØ 1Ø FE 1B
58CE 7A B3 2Ø F9 F3 ØØ ØØ ØØ
58D6 ØØ ØØ ØØ E5 21 ØØ ØØ E5
58DE A7 11 ØØ ØØ ED 52 44 4D
58E6 21 ØØ ØØ Ø9 D1 Ø3 ED B8
58EE 11 F5 FA 21 36 Ø1 Ø1 15
58F6 ØØ D5 ED BØ C9 DB A8 32
58FE F4 FA E6 ØØ D3 A8 CD ØØ
59Ø6 ØØ 3A F4 FA D3 A8 C9 ØØ

Soma total:ØØ6363
```

PROJETO

PARTE IX

Sérgio Duric Calheiros

Na época do surgimento da primeira versão do SCREEN IV, as palavras drive e disquetes ainda não faziam parte do vocabulário dos usuários do MSX. Raras eram as vezes que se via um MSX conectado a um drive, e, mais ainda, sob algum sistema operacional funcionando. Os programas que circulavam pelo país utilizavam somente o cassete, quando não estavam em cartucho.

Os aplicativos que mais fizeram sucesso, e ainda continuam a fazer, são os editores gráficos. Sabendo explorar os recursos do computador, podemos criar telas surpreendentemente complexas, dignas de tê-las em nossas bibliotecas.

Praticamente, a totalidade daqueles editores gráficos disponíveis trabalhavam somente com fita cassete. A transposição de uma tela de um editor para outro era bastante difícil, senão impossível. Ainda mais com os recursos disponíveis na época.

Procurando tornar isso possível, foram incorporadas ao SCREEN IV dois comandos que permitem gravar e recuperar telas que estão na memória de vídeo diretamente para o cassete, que são os comandos SCREEN SAVE e SCREEN LOAD.

Apesar da existência de drives para o MSX atualmente, ainda existem editores que armazenam suas telas em fita cassete. Por isso, longe de se tornarem comandos obsoletos, sem utilidade, os novos comandos ainda podem auxiliar o manuseio de muitos editores gráficos que ainda sobrevivem por aí.

O uso dos comandos é simples, funcionando como qualquer outro comando do BASIC. A sintaxe de cada um deve ser seguida como na tabela abaixo, sendo análogos aos comandos de entrada e saída já existentes no BASIC.

- SCREEN LOAD "<NOME>", <OFFSET> ou
- SCREEN SAVE "<NOME>", <INÍCIO>, <FIM>

No comando SCREEN LOAD, o nome pode ser qualquer seqüência de 6 caracteres. Neste caso, o nome "cas:" não serve para designar um dispositivo de saída, mas apenas um nome como qualquer outro. Para carregar o próximo programa da fita, ou seja, aquele que vem em seguida, caso não saibamos o seu nome, basta dar o comando SCREEN LOAD " ".

O parâmetro offset funciona como nos demais comandos de carregamento, exceto pelo fato do cálculo do endereço final cair na memória de vídeo. Com este parâmetro, poderemos direcionar qualquer bloco binário para a tela, simplesmente escolhendo o offset adequado, de acordo com a posição original do bloco em questão.

Não é necessário dizer que os blocos produzidos pelo comando SCREEN SAVE estão no formato do comando BSAVE. Sabendo como usar os parâmetros offset do BLOAD ou do SCREEN LOAD, pode-se intercambiar os blocos gerados por cada um sem maiores problemas.

A digitação dos comandos não tem segredo algum. Somente fique atento aos endereços dos blocos, já que as chamadas da tabela de jumps não são alteradas, mas sim em outro endereço. Resumindo: Digite o que está apresentado nos lugares indicados nos blocos.

O cronograma previsto para o SCREEN IV está no fim. No próximo número estaremos publicando a última parte do programa, fechando, então, o projeto SCREEN IV. A última parte, como não poderia deixar de ser, é a mais interessante, pois incorpora um dos comandos mais poderosos já inventados para o BASIC: o comando SCREEN OUT. Até lá.

```
BLOCO 1

4378 DD 21 E7 79 C9 C3 A5 19
4380 C3 55 1A 00 00 00 00 00

BLOCO 2

59A1 CD 7B 02 C9 23 DD 21 98
59A9 70 CD 7B 02 2B DD 21 66
59B1 46 CD 7B 02 FE 2C 28 07
59B9 DD 21 55 40 CD 7B 02 DD
59C1 21 55 47 CD 7B 02 ED 53
59C9 51 38 2B DD 21 66 46 CD
59D1 7B 02 FE 3C 20 E2 DD 21
59D9 55 47 CD 7B 02 ED 53 53
59E1 38 22 4F 38 3E D0 DD 21
59E9 25 71 CD 7B 02 AF DD 21
59F1 F8 72 CD 7B 02 2A 51 38
59F9 DD 21 03 70 CD 7B 02 2A
5A01 53 38 DD 21 03 70 CD 7B
5A09 02 21 B2 73 DD 21 03 70
5A11 CD 7B 02 ED 5B 53 38 2A
5A19 51 38 CD 47 1A DB 98 CD
5A21 31 1A CD 84 02 30 03 23
5A29 18 F3 F7 00 F0 00 18 1F
5A31 E5 D5 C5 F5 F7 00 ED 00
5A39 30 07 DD 21 B2 73 CD 7B
5A41 02 F1 C1 D1 E1 C9 7D D3
5A49 99 7C E6 3F D3 99 C9 DD
5A51 21 02 70 C9 23 DD 21 98
5A59 70 CD 7B 02 2B DD 21 66
5A61 46 CD 7B 02 A7 11 00 00
5A69 28 0C FE 2C C2 B9 19 DD
5A71 21 55 47 CD 7B 02 ED 53
5A79 55 38 22 4F 38 0E D0 DD
5A81 21 B8 70 CD 7B 02 DD 21
5A89 E9 72 CD 7B 02 CD EA 1A
5A91 ED 4B 55 38 09 22 51 38
5A99 CD EA 1A ED 4B 55 38 09
5AA1 22 53 38 CD EA 1A 22 BF
5AA9 FC 2A 51 38 ED 5B 53 38
5AB1 CD C9 1A CD D4 1A D3 98
5AB9 CD 84 02 38 03 23 18 F3
5AC1 F7 00 E7 00 21 02 70 C9
5AC9 7D D3 99 7C E6 3F F6 40
5AD1 D3 99 C9 E5 D5 C5 F7 00
5AD9 E4 00 C1 D1 E1 D0 F7 00
5AE1 E7 00 DD 21 B2 73 CD 7B
5AE9 02 CD D4 1A 6F CD D4 1A
5AF1 67 C9 00 00 00 00 00 00

Soma total:009988
```

Softnew

MSX VIDEO GRAPHICS PLUS

Sensacional lançamento da Softnew!
A Softnew coloca a disposição dos usuários do msx, este excelente Editor que irá ajudá-lo na confecção de seus gráficos, com novos formatos e várias outras opções.

MSX DESIGNER

Super Editor Gráfico com 40 fontes de letras, saída para impressora em duplo tamanho com escala de cinza (somente em disco).

MINOS

Finalmente o jogo que você esperava. Um game com mais de 2.300 soluções desenvolvido em Turbo Pascal com grande variedade de movimentos. Acompanha manual altamente explicativo. Você pode inclusive salvar situação em que parou no jogo e continuar no dia seguinte. Poucos que já jogaram o Minos conseguiram entre 10 e 15 soluções. Tente você vencer este desafio.

MULTICOPY

Enfim o Copiador que você esperava! Realiza cópias Disco/Disco; Disco/Fita; Fita/Fita; Fita/Disco; Disco/Fita automático; Diretório na impressora; Formata e coloca o diretório do seu disco em ordem alfabética, dispondo ainda de mais de sete opções de velocidade para gravação em fita, além de muitos outros recursos.

SPRITE WRITE

A Softnew mais uma vez valoriza o programador nacional, lançando um software totalmente desenvolvido por nós possibilitando a você mais uma opção a nível de programa.
O mesmo destina-se a alterações e criação de sprites possibilitando a você modificações em jogos ou mesmo a inserção de novos sprites em seus programas.

REVENDEDORES SOFTNEW

SÃO PAULO SP: FILCRIL (011) 220-3833 MSX Informática (011) 872-0737 Ressom (011) 229 7031 Ectron (011) 290 7266 Redi-Universoft (011) 825-5240 Datatec (011) 571 7083 Tall Comunicações (011) 914 2844 Misc (011) 36-3226. Champion (011) 65 2030 Casa do MSX (011) 533 2351. Star Computer (011) 280-4722. MCC Informática (011) 262-1870. Paulisoft Informática (011) 37-1814 Game Of Time (011) 581-2739 Audicomp (011) 267-3581 - Benny (011) 570-1555

SÃO PAULO - INTERIOR, SÃO BERNARDO DO CAMPO Microspend (011) 448-6288 (Golden Shopping). SANTOS: Data Market (0132) 35 7500 / Taco Soft (0132) 33-2037. CAMPINAS Microdata Informática (0192) 31-9766 VALINHOS: Sede Informática (0192) 71-3331. RIBEIRÃO PRETO: ALS (016) 636 5379

OUTROS ESTADOS: RIO DE JANEIRO RJ: Riosoft Informática (021) 264 3720. Casa da Informática (021) 231-1592 / (021) 521 2844 Informtelles (021) 751-5078 Shalon (021) 796-3847 BRASÍLIA - DF: Intersoft (061) 244 5728 CEARÁ CE: Top Data Informática (085) 239 1618 A&A Software - Caixa Postal 201 - Sapiranga - RS PORTO ALEGRE - RS Prólogos Informática (0512) 22-5803 BAHIA BA: Soft e Periféricos (071) 240-3128 MINAS GERAIS - MG: Ror Soft (033) 621-2713 / (033) 621 2122. SANTA CATARINA - SC: Megabyte (0482) 23-5010. ESPÍRITO SANTO ES: Vitória Video Center (027) 235 1996 BELEM - PA: Compubel (091) 223 6319.

SUPER NOVIDADE

Conheça nossa sensacional coleção de games em nossos revendedores. Temos 10 super games com mais de 25 modelos a sua disposição.

JOGOS

A emoção e a aventura esperam por você na Softnew!
São mais de 2.000 jogos, além dos mais recentes lançamentos do mercado.
A Softnew também é lazer e entretenimento.

ESTA É A SUA GRANDE CHANCE!

Se você tem um software criado por você, procure-nos. Nós incrementamos, legalizamos e promovemos o seu software.
É a Softnew em busca de novos talentos na informática.

SE VOCÊ QUER SER REVENDEDOR SOFTNEW, PROCURE NOSSO DISTRIBUIDOR EXCLUSIVO ACS - (011) 910-9923

SOFTNEW INFORMÁTICA LTDA.
Rua Miguel Maldonado, 173 — Jd. São Bento
CEP 02524 — São Paulo — SP
Tel.: (011) 266-2902

LADY 80 - A PEQUENA NOTÁVEL

Recebemos, em nossa redação, para análise, a impressora Lady 80, que logo recebeu o apelido de "A pequena notável", devido ao seu tamanho reduzido, inúmeros recursos e qualidade de impressão.
A Lady 80 foi desenvolvida para atender o mercado MSX e PC. Se a versão for ABNT-MSX, todos os caracteres do MSX, inclusive os caracteres gráficos e acentuados, estarão disponíveis.

EMBALAGEM

A Lady 80 vem de fabrica em uma embalagem de cartão, sendo a impressora acondicionada em dois suportes de isopor, que a protegem de eventuais impactos que possa sofrer durante o transporte. Tanto a impressora como o tracionador e o cartucho da fita são embalados em sacos plásticos.

O conjunto é bom e serve perfeitamente à sua finalidade, evitando surpresas desagradáveis na hora de abrir a caixa.

MANUAL DE OPERAÇÃO

O manual que acompanha a Lady-80 é excelente, tanto na apresentação como no conteúdo. Ao lê-lo, o usuário, mais uma vez, tem a certeza de que adquiriu um produto de excelente qualidade e que não terá dificuldades na sua instalação e posterior uso.

Da forma que é apresentado, o manual pode ser utilizado como uma fonte de consulta, sempre que for necessário.

Além do manual, o usuário recebe um "Resumo de Comandos", ou seja, um cartão de consulta, impresso em papel plastificado e cuja a função é a de facilitar uma consulta rápida.

Sem dúvida alguma, a Elgin tem uma preocupação muito grande com o usuário de seus produtos e só merece elogios pelo manual que fornece.

INSTALAÇÃO

A instalação do equipamento é simples e não apresenta dificuldade alguma para a ligação dos cabos, colocação do tracionador e fita.

A colocação da fita de impressão, que em algumas impressoras é uma tarefa que exige do usuário paciência e a lavagem das mãos, é extremamente simples e dispensa o contato com a fita. O sistema utilizado é bem pensado e mais uma vez o usuário sai ganhando.

CHAVES DE CONFIGURAÇÃO

A Lady 80 possui dois conjuntos de chaves de programação, que possibilitam ajustes por hardware.

Através destas chaves é possível selecionar:

– Tipo de interface e paridade

– Transmissão de X-ON/X-OFF
– Velocidade de transmissão
– Número de Bits do caracter
– Salto de fim de página
– Tipo de impressão (QC ou normal)
– Tamanho do formulário
– Formato de impressão
– Conjunto de caracleres
– Zero cortado
– Buffer de dados
– Retorno do carro com ou sem avanço de linha
– Entrada de dados

A configuração destas chaves só é necessaria quando se faz a inslalação do equipamento ou quando se deseja uma aplicação mais específica.

A configuração de fábrica irá atender à grande maioria dos usuários, que não necessitarão fazer uso de mais este recurso oferecido pela impressora. Contudo, é importante que o usuário saiba da sua existência, para que possa tirar o máximo proveito de seu equipamento.

Alguns dos recursos disponíveis através do conjunto de chaves também podem ser acessados via software.

Como em todo equipamento, só a leitura do manual poderá ajudar o usuário a decidir a melhor forma de utilização.

CARACTERÍSTICAS TÉCNICAS

São as seguintes as características técnicas, fornecidas pelo fabricante.

Método de impressão: Matricial
Cabeça de impressão: 9 Agulhas
Velocidade de impressão: 100 caracteres por segundo
25 caracteres por segundo em qualidade carta

Vida útil da cabeça de impressão: 50.000.000 caracteres
Extensão da linha: 8 polegadas (203,2 mm)
Cópias: 3 vias, incluindo o original
Alimentação do papel: fricção, tração
Espaçamento da linha: 1/6 de polegada, 1/8 de polegada ou salto programado de múltiplos de 1/216 de polegada
Largura da folha de impressão: Folha solta: formado A4 210,8 mm formato carta 216,0 mm
Formulário contínuo de 101,6 a 254 mm

Tipo de fita: Largura 8 mm – Comprimento 10 m (cor: preta)
Vida de 1.000.000 de caracteres
MTBF: 4.000 horas (excluindo a vida da cabeça de impressão)
Condições Ambientais: Temperatura de 5 a 35 graus Celsius
80% de umidade do ar (máximo)

Tensão: 120 V alternada, tolerância 10%
220 V (opcional)
60 Hz
Potência consumida: 30 W
Nível de ruído: Menor que 60 dB
Dimensões: 334 mm de comprimento
195 mm de profundidade
70 mm de altura
Peso: 3,5 kg
Interface: Paralela Cetronics
RS-232 serial (opcional)
Direção de Impressão: **IMPRESSÃO BIDIRECIONAL**
Normal/Expandida
Comprimida/Enfatizada/Proporcional
Elite/Sobre e subscrita
Impressão Unidirecional:
IMPRESSÃO UNIDIRECIONAL
Modo gráfico (esquerda-direita)
Qualidade de Carta (QC)

Quantidades de Caracteres Diferentes: 191 caracteres no conjunto ABNT, com as seguintes variações para os conjuntos Internacionais
1 no inglês, 8 no alemão
8 no francês, 6 no dinamarquês I
11 no sueco, 7 no italiano
6 no espanhol, 1 no japonês
11 no norueguês, 10 no dinamarquês II
8 no português,

Largura do Caracter:

	Padrão		Expandido		Sobre Subscrito	
	Altura (mm)	Largura (mm)	Altura (mm)	Largura (mm)	Altura (mm)	Largura (mm)
Normal	2,4	2,0	2,4	4,0	1,2	2,0
Elite	2,4	1,7	2,4	3,3	1,2	1,7
Comprimido	2,4	1,2	2,4	2,4	1,2	1,2

Formação de Caracter: Caracter normal 9x9
QC 18x20

	60	72	80	90	120	144	240
Modo Gráfico	60	72	80	90	120	144	240
Densidade	1				2		4

Densidade: Horizontal 1/60 de polegada (normal)
1/72 de polegada (elite)
Vertical 1/72 de polegada

Número de Caracteres por linha (Número de Caracteres por polegada)

Normal	Padrão:	80 (10 cpi)
	Expandido:	40 (5 cpi)
	Enfatizado:	80 (10 cpi)
Elite	Padrão:	96 (12 cpi)
	Expandido:	48 (6 cpi)
Comprimido	Padrão:	136 (17 cpi)
	Expandido:	68 (8,5 cpi)
Elite Comprimido	Padrão:	160 (20 cpi)
	Expandido:	80 (10 cpi)

TRACIONADOR DE PAPEL

Colocando o tracionador de papel, pode-se utilizar formulários contínuos.

Na maioria das impressoras, o usuário tem que optar: ou usa o tracionador e imprime em formulário contínuo ou retira o tracionador e usa folha solta.

No caso da Lady 80, o usuário pode utilizar folha solta, mesmo estando com o tracionador acoplado, bastando tirar o formulário contínuo. Quem utiliza formulário contínuo para imprimir listagens e papel carta para uma correspondência, sabe que só o fato de não ter que ficar colocando e tirando o tracionador representa uma enorme vantagem.

COMBINAÇÃO DOS MODOS DE IMPRESSÃO

A Lady 80 opera com quatro modos de impressão básicos e 5 subsidiários, além da impressão em qualidade carta. No modo básico temos elite, proporcional, enfatizado e normal. E, no subsidiário, estão disponíveis o modo comprimido, sobre/subscrito, negrito, expandido e sublinhado.

Os modos básicos de impressão não podem ser selecionados juntos, ou seja, não podemos selecionar elite e proporcional ao mesmo tempo, mas podem ser selecionados em conjunto com os modos subsidiários.

Deste modo, por exemplo, podemos selecionar a impressora para imprimir em normal, comprimido, negrito, expandido e sublinhado.

A variação e combinação de vários tipos de impressão é útil quando se deseja imprimir um texto e esbanjar recursos, para melhorar a apresentação.

O editor de textos Astex, analisado neste número de CPU, pode fazer uso de alguns destes recursos.

MODO PROPORCIONAL

Um outro recurso presente nesta impressora é a impressão proporcional, ou seja, o espaço entre os caracteres é proporcional à largura dos mesmos.

MODO GRÁFICO

A impressão de gráficos na Lady 80 foi de boa qualidade. Vários editores gráficos foram testados, sem ter tido ocorrido qualquer tipo de problema, imprimindo os gráficos perfeitamente, com boa resolução e uma excelente definição.

GARANTIA

A Elgin garante a Lady 80 por um prazo de 12 meses, a contar da data de compra, contra

qualquer defeito de fabricação ou material, observadas as condições de uso.

A rede de assistência técnica possui postos nas principais capitais, cobrindo 24 cidades, o que pode ser considerado excelente, em se tratando de um equipamento de informática.

Junto com a impressora, é fornecido uma relação de oficinas autorizadas, onde se pode encontrar todas as informações necessárias para um pronto atendimento.

DESIGN

Um dos pontos que chama a atenção na Lady-80 é extamente o seu tamanho reduzido (334 x 195 x 70 mm), com design moderno e leve.

O posicionamento das teclas e Leds indicativos facilita a operação e visualização.

trabalhos que produz têm sido todos de excelente qualidade. As listagens deste número, por exemplo, foram impressas na Lady-80.

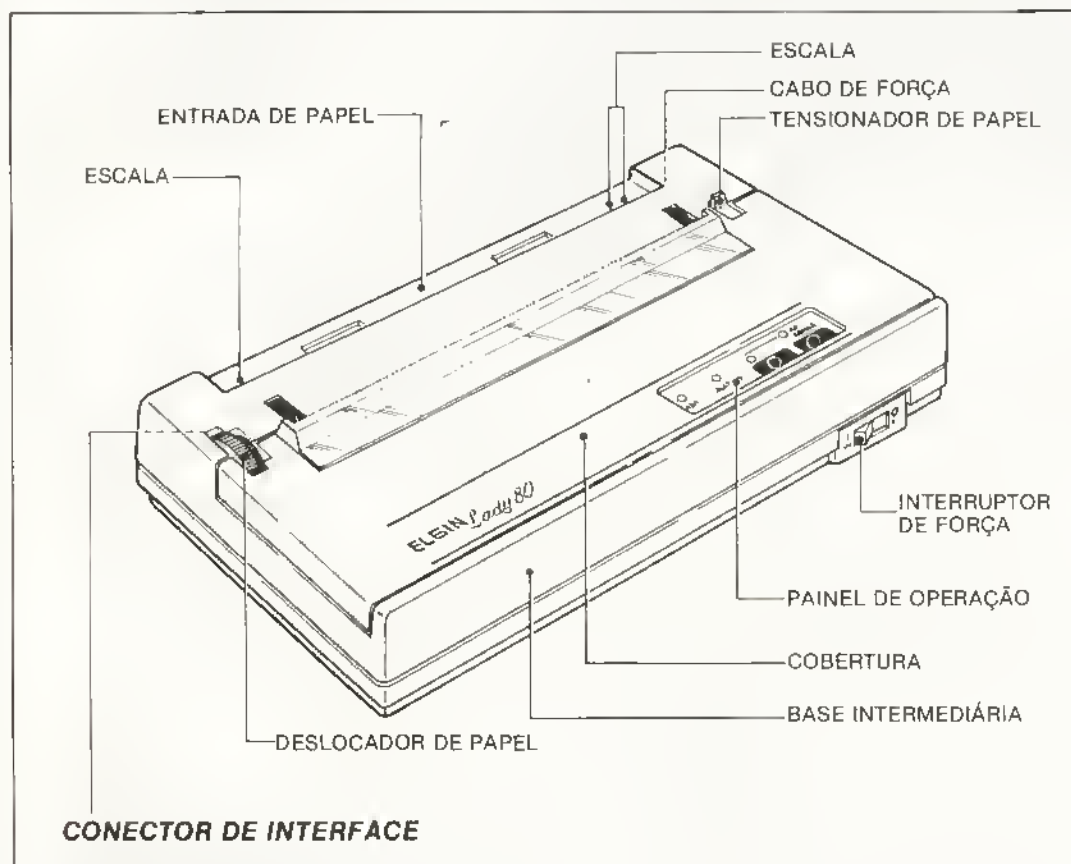
Nos trabalhos mais demorados, como a impressão de cadastro de assinantes e etiquetas de endereçamento, que levaram até 5 horas de impressão contínua sem intervalo, não apresentou qualquer tipo de problema ou mesmo falha de impressão.

CONSIDERAÇÕES FINAIS

A Lady 80 é uma impressora que impressiona bastante.

A maior parte dos recursos que oferece ao usuário só estão disponíveis em impressoras de custo bem elevado.

A velocidade de impressão, de 100 caracteres por segundo, pode parecer, em uma análise superficial e imediata, um tanto lenta mas, para o tipo de usuário para o qual se destina e foi desenvolvida, é mais do que suficiente, efetuando um bom trabalho num prazo de tempo perfeitamente aceitável.

No painel de comandos, faz falta a tecla para avanço de página e este foi o único ponto que achei falho, pois é extremamente útil quando

DESEMPENHO

Desde que chegou à nossa redação, temos utilizado a Lady 80 para várias tarefas, como impressão de relatórios, cartas para os leitores e todo o trabalho que necessitava ser impresso.

Em pouco tempo, todos aprenderam a operá-la, tarefa esta extremamente simples, e os

se trabalha com formulário contínuo e não se tem acesso ao Basic.

Sem dúvida alguma, a Lady 80 é uma excelente impressora, onde não faltam recursos e possibilidades. Atende a uma gama variada de usuários e presta-se a vários tipos de serviços.

Sua qualidade de impressão, em modo qualidade carta, necessário quando desejamos imprimir um texto mais apresentável, é excelente.

O nome Elgin, por si só, já inspira confiança e significa vários anos de experiência e bons produtos. A Lady 80 é mais um deles, que confirma o respeito que a empresa conquistou junto ao público consumidor.

```
39:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmno
ª«¬-®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ ØÙÚÛÜÝÞßàá
ûüýþ
,-./0123456789:;<=>?@ABCDEFG
TUVWXYZ[\]^_'abcdefghijklmno
39:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmno
ª«¬-®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ ØÙÚÛÜÝÞßàá
ûüýþ
39:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmno
ª«¬-®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ ØÙÚÛÜÝÞßàá
ûüýþ
JKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~
39:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmno
=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~
/0123456789:;<=>?@ABCDEFGHIJKLMNO
^_'abcdefghijklmnopqrstuvwxyz{|}~
=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~
QRSTUVWXYZ[\]^_'abcdefghijklmnopqrstuvwxyz{|}~
39:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'abcdefghijklmno
ûüýþ
,-./0123456789:;<=>?@ABCDEFG
TUVWXYZ[\]^_'abcdefghijklmno
```

Amostra de Impressão

O NOVO PROCESSADOR DE TEXTO ASTEX:

Paulo Roberto Pinheiro Elias

Havia apenas uns quinze a vinte dias que eu havia adquirido a nova placa de 80 colunas da Gradiente, quando recebi um convite do nosso Editor Gonçalo Murteira, para fazer uma análise do novo Processador de Textos ASTEX, desenvolvido por Fernando A. Santor Jr., programador de São Paulo, com quem mantive recente contato por carta, a respeito deste Editor.

O ASTEX vem embutido na placa CT-80E, que usa o mesmo chip de vídeo do MSX 2.0 e possui absoluta integração com os comandos do BASIC (WIDTH, VPOKE, etc.) e do DOS (MODE) relativos às características do terminal de vídeo do MSX. Assim, o ASTEX faz uso pleno das técnicas de redefinição de caracteres que tornam possíveis o aparecimento de sublinhados, itálicos, negritos e caracteres em vídeo reverso simulado, dando ao aplicativo um aspecto extremamente atraente e de grande apelo visual. O objetivo destas redefinições é o de exibir, na tela do computador, o aspecto final do texto na impressora.

Antes de falar sobre o ASTEX, gostaria de comentar que esta nova placa de 80 colunas é o melhor periférico deste tipo disponível no momento, não só por respeitar os comandos do computador como também pela sua qualidade de imagem, no que concerne à resolução, estabilidade (não há oscilações significativas de sincronismo) e o aspecto final dos caracteres em tela de texto de 80 colunas. O encaixe da placa num dos slots externos é suficientemente firme para sustentar o cartucho sem problemas. Frustrante, porém, é a ausência de sinal de vídeo composto na saída da placa, o que impede que a mesma possa ser ligada num monitor colorido que possua esta entrada. Dependendo da composição de cores escolhida pelo usuário, a imagem gerada pode se tornar borrada em um monitor de fósforo verde, dificultando a visualização dos caracteres. Eu sinceramente confesso que não entendo a filosofia que norteia os projetistas da Gradiente.

Pior do que isso, entretanto, é o Manual de Instruções que acompanha o produto. Fiquei de uma certa forma perplexo com a má qualidade do Manual, visto que foi elaborado pela competente equipe da Editora Aleph, dedicada ao padrão MSX. Houve uma série de descuidos que a Editora deveria corrigir para o bem de seus leitores. Um deles, por exemplo, eu descobri logo quando manuseei o Manual pela primeira vez: as folhas se despencaram quase completamente nas minhas mãos.

Outro problema sério deste Manual é a parcimônia de informações contidas no seu interior. Isto é bastante prejudicial aos usuários em potencial do ASTEX. Felizmente, o autor do programa se propõe a socorrer por carta aqueles que, como eu, desejariam se informar melhor sobre o Editor. Eu sei perfeitamente que é difícil editar um livro isento de erros ou problemas, mas o caso aqui é não só a ausência quase absoluta de listagens, como irei comentar adiante, mas também de orientações erradas ao leitor. O ASTEX é um aplicativo suficientemente poderoso para justificar um livro capaz de permitir a exploração de seus recursos, mas não um Manual como este.

RECURSOS DE EDIÇÃO DO ASTEX:

O ASTEX segue os padrões dos mais modernos e bem aceitos Processadores de Texto da linha IBM-PC, tais como o WORD da MICROSOFT ou o FRAMEWORK (sistema integrado de aplicativos profissionais bastante apreciado pelos seus usuários).

Embora não chegue a ser um software de editoração eletrônica ("desktop publishing"), o ASTEX permite a colunagem do TEXTO, a composição de páginas de livro (com o espaçamento alternativo do texto na folha impressa, de acordo com a numeração das páginas) e a sua visualização em modo gráfico do aspecto final do documento que será impresso. No item "colunagem", este recurso é bastante importante, pois, durante a edição, o usuário não consegue enxergar na tela de digitação o aspecto final das colunas. Não obstante, poderá se guiar facilmente pela linha de status inferior, que indica as coordenadas exatas do cursor, incluindo, no caso, a coluna que está sendo digitada num determinado momento.

Como todos os Processadores de Texto anteriormente citados, o ASTEX exige do usuário uma definição do parágrafo que servirá de base para a composição do texto. O programa entra com um default propositalmente sem justificação à direita, de tal maneira que o usuário poderá digitar todo o texto de uma só vez sem se preocupar com a formatação e só depois redefinir cada parágrafo individualmente, de acordo com o acabamento que deseja do texto. Todas as formatações e reformatações de parágrafo são automáticas, mas o programa não é capaz de criar, destruir ou reconstruir as inevitáveis hifenações de palavras, tal como os softwares mais atuais de editoração eletrônica do PC. A ausência deste recurso em muito prejudica a edição do texto, pois no caso de alterações em um parágrafo contendo palavras hifenadas, o usuário deverá ele próprio redigitar as linhas que continham hifenações. Uma escapatória para esta penalização é a digitação do chamado "soft-hyphen" (chamado de "hífen não requerido" pelo Manual e de "hífen opcional" pelo autor do programa). Este tipo de hífen pode ser digitado em qualquer posição do texto, mas só será efetivado na impressão quando presente no fim de uma linha. Mesmo assim, o digitador se obrigará a rehifenar manualmente, nos casos de reforma de algum trecho do parágrafo

Para aqueles que porventura já usam o ASTEX, vou contar como resolvi este problema com um mínimo de mão-de-obra. Sendo a hifenação do texto imprescindível para homogeneizar o texto e economizar espaço no arquivo, no caso de alguma reforma de parágrafo, o usuário deverá fazer o seguinte: acionar o modo de inserção automática (tecla INSERT), levar o cursor para o fim da primeira linha desarrumada e digitar o texto imediatamente a seguir. Você notará que as palavras à esquerda recuarão, e os espaços em branco excedentes serão automaticamente preenchidos. Preste bastante atenção na tela, pois quando não sobrarem mais brancos excedentes disponíveis, o cursor se deslocará para a próxima linha. Delete então os caracteres da próxima palavra que estiverem re-

PERIFÉRICOS

TUDO PARA MSX

DRIVE
5 1/4"

CANETA ÓTICA

3 1/2

MODENS

TECLADOS
RS 232
TELEX PARA MSX E PC

EXPANSOR
DE SLOTS

PROGRAMAS
NEMESIS
PAULISOFT
SOFTNEW
XSW
PRÁTICA
DISCOVERY
YOUNGSOFT
CIBERTRON

LANÇAMENTO
COMPILADOR MOZART
CIBERTRON

DISKETES

REMETEMOS PARA TODO
O BRASIL VIA SEDEX

ENTREGA IMEDIATA

ÁGUIA INFORMÁTICA LTDA.

Av. N.S. de Copacabana, 605/804 – Copacabana
Telex: 21 21 717 KPUR – Tel.: (021) 235 3541
Rio de Janeiro – RJ – CEP 22040

duntantes (tecla DELETE) e passe ao fim da linha para repetir esta operação, até que o parágrato termine. O salto do cursor para o fim de cada linha é facitmente conseguido teclando-se CONTROL + SETA PARA A DIREITA. Importante: todos os hífens digitados pelo usuário no fim de cada linha são considerados "quebras de palavra" e assim promoverão a passagem do restante da palavra hifenada automaticamente para a tinha seguinte. Quando um parágralo é formalado, o espaço residual entre as palavras é mostrado na lela pelo alongamento do cursor, dentro do texto, à medida em que este se posiciona entre as palavras espaçadas.

Um recurso que o Manual não conta e que ajuda a driblar a formalação imposta peto Editor reside na possibilidade de se digitar SHIFT + ESPAÇO, obtendo um minúsculo ponto entre as patavras. Fazendo isso, pode-se impedir que este espaço seja manipulado pelo relormalador automático de parágratos. Este recurso é particularmente útil quando se deseja lormatar uma palavra composta de modo a que ela saia sempre com o mesmo espaçamento, independente da posição que ela ocupe na linha de texto É como se colássemos as palavras uma na oulra, através destes minúsculos pontos. No WordStar, este recurso é chamado de "espaço inseparável" ("non-break space").

No meu entender, o usuário do ASTEX deve providenciar de imediato as **formatações de parágrafo** desejadas. Antes, porém, deverá definir as **DIMENSÕES DO DOCUMENTO** a ser impresso (sobre isto falaremos mais adiante). Como no Manual os valores default de formatações pré-existentes não estão listados, é altamenle conveniente copiá-los em um papel à parte e exercitar um pouco de digitação com cada um deles para sentir o que cada formatação é capaz de fazer com o lexto digitado. Depois, torna-se interessante definir tipos de parágratos novos ou então aproveitar parte dos já existentes Ao tormatar a página a ser impressa (opção página e depois Dimensões/margens), o Editor se ajustará automaticamente às características de digitação de cada parágrafo, mostrando na lela o exato espaço disponível para o usuário. Por esle motivo, é importanle que isto seja feito em primeiro lugar, principalmente se você tem o hábito, como eu, de hilenar o texto. Normalmente, será preciso ajustar o número de colunas relalivo ao tamanho do tormulário, caso a impressora disponível seja de 80 colunas, pois o ASTEX vem com um detault de 85 colunas/linha. Depois, se desejado, pode-se alterar as margens esquerda e direita, neste caso delimitando-se **o espaço máximo** que o texto poderá ocupar no papel. Se as margens esquerda e direita de um parágrafo forem zero, o Edilor mostrará na tela de digitação estes valores, caso contrário, descontará os valores estabelecidos no formato de parágrafo adolado, estreitando o espaço para a digitação do texto. Por exemplo: estabelecendo-se 80 colunas como **largura** do formulário e delimitando-se o texto para uma margem esquerda de 5 colunas e direita de 3 colunas, o máximo de texto que caberá numa linha será de 72 colunas. Quando um parágrato for definido para começar a partir da 5ª coluna, isto corresponderá à 10ª coluna dentro do formulário.

Uma característica muito inleressante do ASTEX é o reconhecimento por ele próprio das lormatações de parágrafo impostas pelo usuário ao Editor Isto signitica que, a qualquer momento que o cursor passar por um delerminado parágrafo, as informações de tela referir-se-ão à sua formatação e a sua Edição obedecerá a este lormato, mesmo que se esteja no momento trabalhando com outra atribuição em outro local do texto.

A digitação no ASTEX é enorme-mente facilitada pelas informações de tela, que podem, ao gosto do usuário, serem suprimidas, o que aliás é lortemente contra-indicado, a não ser por motivos muito especiais. Não consegui ver nenhuma informação redundante na tela, muito ao contrário: em cima, uma régrua numerada com um cursor indica ao usuário, através de colchetes, os locais exatos

Figura 1

```
Drive:B              CPU9        .TXT                   544

02  A6  52  23  0D  0A  02  23            .&R#...#
4F  20  4E  4F  56  4F  20  50            O NOVO P
52  4F  43  45  53  53  41  44            ROCESSAD
4F  52  20  44  45  20  54  45            OR DE TE
58  54  4F  20  41  53  54  45            XTO ASTE
58  3A  0D  0A  02  26  50  61            X:...&Pa
75  6C  6F  20  52  6F  62  65            ulo Robe
72  74  6F  20  50  69  6E  68            rto Pinh
65  69  72  6F  20  45  6C  69            eiro Eli
61  73  0D  0A  02  26  0D  0A            as...&..
02  26  0D  0A  02  21  48  61            .&...!Ha
76  69  61  20  61  70  65  6E            via apen
61  73  20  75  6E  73  20  71            as uns q
75  69  6E  7A  65  20  61  20            uinze a
76  69  6E  74  65  20  64  69            vinte di
61  73  20  71  75  65  20  65            as que e
```

```
Drive:B              CPU9        .TXT                   661

6E  64  65  6E  73  61  64  61            ndensada
20  65  20  44  20  2D  20  69             e D - i
6D  70  72  65  73  73  B1  6F            mpressão
20  65  78  70  61  6E  64  69             expandi
64  61  2E  1A  AA  55  01  3C            da..¬U.<
01  03  03  01  01  40  08  08            .....@..
00  00  00  00  87  00  00  00            ....ç...
0D  41  52  47  4E  53  00  01            .ARGNS..
00  80  00  C0  00  D0  41  52            .Ä....AR
41  47  31  00  02  00  80  00            AG1...Ä.
C0  00  50  41  52  41  47  32            ..PARAG2
01  03  00  87  00  A8  00  04            ...ç.¿..
54  4C  20  31  20  22  04  04            TL 1 "..
83  00  C0  00  45  45  3D  34            â...EE=4
20  20  00  05  00  87  00  40              ...ç.@
00  43  41  42  45  43  41  00            .CABECA.
```

Formatação de gravação de um arquivo-texto digitado no ASTEX, obtida com o auxílio do utilitário HELLO. As telas exibidas referem-se a primeira e última páginas do arquivo, onde se pode notar alguns Bytes de identificação dos parágrafos e os códigos de delinição desses parágratos, gravados após o Byte de Fim-de-Arquivo (&HIA).

onde iniciam e terminam as colunas de texto. As tabulações são marcadas com asteriscos, mas estes sinais podem ser mudados, a critério do usuário. Um sinal de 2 traços(=)marca o ponto parágrafo na 1ª linha, indicada pela formatação de parágrafo escolhida. Na linha de status, situada em baixo, aparecem informações como página, linha e coluna (posição do cursor) e a outra coluna referente à posição do texto colunado, quando cabível. Ao lado, outras informações como fontes de caracteres, formatação de parágrafo e memória livre estarão disponíveis. Acima da linha status, pode-se visualizar o quanto de memória já foi gasta, através do preenchimento de uma linha contínua, onde, na sua extremidade, pisca um micro-cursor, o qual se desloca pela linha mostrando a posição de memória relativa á posição atual do cursor. Finalmente, no canto direito desta mesma linha, aparece a palavra ASTEX em destaque e durante a impressão, surgirá, no mesmo estilo, mais abaixo, o nome do filtro de impressão.

Cada parágrafo editado no ASTEX possui um header invisível para o usuário. Este header, uma vez gravado no disco, permite que as características de formatação estipuladas pelo usuário possam ser resgatadas numa futura edição. Se o ASTEX for utilizado para a edição de programa fonte ou qualquer outro tipo de arquivo desta natureza, o usuário deverá optar pela inibição da gravação desta formatação, quando for salvá-lo. Todas as formatações de texto poderão ser reaproveitadas para a geração de outro arquivo-texto, através da opção de limpeza exclusiva do texto. Na Figura 1, mostra-se a varredura do conteúdo de um arquivo digitado no ASTEX, onde se pode notar que antes do texto aparecem alguns Bytes que referenciam o parágrafo e após o fim do texto, sinalizado pelo Byte & H1A (End-of-File), são gravadas as características dos parágrafos formatados.

CORRESPONDÊNCIA ENTRE O TEXTO DIGITADO E A IMPRESSÃO EM PAPEL:

Todos os caracteres, palavras ou blocos de texto, que forem individualmente configurados para impressão em negrito, sublinhado, itálico, indice ou expoente, aparecerão gloriosamente com esta formatação na própria tela do computador. Mas espere. . . antes que você tenha um orgasmo precoce, devo esclarecer o seguinte: o ASTEX não simula estes caracteres graficamente na impressora, como fazem alguns Processadores de Texto do PC. É necessário que a sua impressora obedeça aos comandos que permitem a obtenção destas fontes. Se não for este o caso, o usuário poderá utilizar os labels do filtro de impressão correspondentes a estas fontes e enviar códigos de controle distintos dos originais, caso haja alguma outra fonte disponível. Para quebrar um galho, o próprio ASTEX simula, ele mesmo, alguns tipos de fonte. Por exemplo, o negrito é conseguido passando a cabeça de impressão várias vezes pelo mesmo caracter (recurso idêntico ao do WordStar). O sublinhado é obtido de forma idêntica, exceto que a cabeça de impressão retrocede para imprimir um traço embaixo dos outros caracteres.

VOLUME DE TEXTO DIGITADO:

O volume de texto digitado através do Processador é limitado pelo fato do programa trabalhar o arquivo-texto totalmente na memória do micro. Cerca de 40 Kbytes estão abertos para esta finalidade. Quando o usuário tem necessidade de trabalhar com arquivos mais extensos, o ideal seria o programa criar um arquivo temporário no disco, que não é o caso. Sendo assim, a única solução é dividir o trabalho em vários arquivos. Por outro lado, existe uma vantagem implícita neste sistema, pois a velocidade de manipulação do texto é incomensuravelmente maior. De fato, é possível realizar algumas operações no Editor, principalmente àquelas que se relacionam ao deslocamento do cursor e ao rolamento ("scrolling") do texto, quase instantaneamente.

RECURSOS DE GRAVAÇÃO DO ARQUIVO-TEXTO:

Depois que alguém se dá ao esforço de digitar um longo trabalho, principalmente se estiver envolvida alguma creatividade na formulação do texto, o que mais se deseja saber é como salvá-lo. O ASTEX permite a gravação do arquivo-texto em cassete ou disco, sendo que para os usuários sofredores do primeiro periférico, é oferecida a opção de gravar em 2.400 bauds, o que, aliás, é

altamente desaconselhável.

O ASTEX não faz backup automático do arquivo lido do disco que está sendo reeditado, um esquecimento do autor, que sugerimos seja corrigido na próxima Versão. Em contrapartida, avisa ao usuário sobre a existência prévia do arquivo no disco, solicitando a sobreescrita. Teclando-se a SETA PARA CIMA, pode-se visualizar e manipular o diretório do disco no drive alocado, bem como saber a memória livre do mesmo.

Talvez um dos recursos mais interessantes do ASTEX para a gravação de arquivos-texto e sequer chamado a atenção do leitor pelo Manual, é a possibilidade de se salvar um arquivo-texto no disco dentro do formato de impressão, o que significa passá-lo pelo filtro de impressão, que neste caso, se torna um filtro de gravação. Se o usuário criar um filtro de impressão com trocas de códigos adequadas, poderá passar seus arquivos diretamente para outro Editor ou até outro sistema ou computador. Neste particular, é importante mencionar que existe a possibilidade de se gravar o arquivo-texto sem a formatação dos parágrafos definidos no Editor, o que significa salvar o arquivo no chamado "formato ASCII puro", onde constam apenas os caracteres desta tabela, sem os códigos de controle, ou seja, exclusivamente texto.

OPÇÕES DE IMPRESSÃO NO ASTEX:

Além da possibilidade acima descrita, o ASTEX possui recursos de impressão em formulários ou folha solta, ao gosto de cada um. Mas, no caso, o mais importante, é que o Editor permite a criação e compilação de um filtro de impressão pelo usuário, nos casos onde os filtros residentes não atendam às especificações da impressora ou à qualquer outra finalidade. Este é talvez um dos pontos mais fracos do Manual de Instruções, que, além de omitir as listagens dos filtros residentes, ainda desorienta o leitor na hora de construir seu próprio filtro, porque não existem explicações convincentes sobre os procedimentos por ele sugeridos. Por exemplo: o Manual diz que o usuário poderá aproveitar partes dos filtros residentes e só alterar os labels que interessam. Mas, na realidade, se fizermos alguns testes por conta própria, iremos verificar que algumas partes do filtro não podem ser omitidas, ao contrário do que o Manual afirma, enquanto que certos labels, quando omitidos, ou não prejudicarão a impressão ou serão automaticamente substituídos por recursos do próprio Editor, como mencionei anteriormente, nos casos das simulações de negrito e sublinhado na impressora. Isto leva o usuário a uma certa confusão, pois não se sabe "a priori" quais os labels que, se omitidos, não trarão prejuízo ao funcionamento do filtro.

Outro ponto sem explicação no Manual refere-se a digitar um determinado label apenas teclando RETURN no lugar dos códigos. O que isto significa??? Um dos labels onde esta digitação é sugerida é o de "reset" da impressora (de nº 11). Eu experimentei omiti-lo da listagem e aparentemente nada mudou. No entanto, se o mesmo for feito com outros labels, o Editor não imprime nada direito, donde se conclui que esta digitação anula algum valor contido anteriormente naquele label, efeito semelhante à omiti-lo da digitação do filtro.

DICAS PARA A CONSTRUÇÃO DE UM FILTRO DE IMPRESSÃO:

Se você já é ou vai ser um usuário do ASTEX e pretende montar seu próprio filtro, talvez ajude um pouco o que vou relatar a seguir.

Note que um filtro de impressão possui duas partes distintas e não excludentes: a primeira, relativa aos códigos de controle necessários ao Editor e a serem enviados á impressora; e a segunda, relativa aos códigos dos caracteres que necessitam ser trocados ou compostos por algum artifício gráfico, durante a impressão. A única coisa comum a estas duas partes são os labels #00: e #99:, que correspondem ao início e fim do filtro, respectivamente. Assim, mesmo que só seja necessária a implementação da segunda parte do filtro, a primeira terá que ser digitada com todos os seus códigos, exceto alguns labels referentes a tipos de caracteres como, por exemplo, negrito, sublinhado, etc., que são aqueles que o próprio Editor pode simular.

Na primeira parte do filtro, o label #00: define em qual tabela residente (MSX ou ABNT) se baseará seu filtro. Esta escolha só terá relevância **SE** o filtro ficar resumido à primeira parte, sem redefinição dos códigos da tabela de caracteres, pois nos casos onde o usuário redefine códigos, ele o faz somente dos caracteres acentuados. Os outros códigos são comuns a qualquer uma das tabelas escolhidas, tornando assim esta escolha irrelevante. Para tentar deixar isto bem claro, vamos dar o seguinte exemplo: o usuário possui uma impressora que segue uma tabela MSX ou ABNT, mas os códigos de controle para as fontes de caracteres são diferentes. Neste caso, a simples opção do label #00: definirá os códigos dos caracteres acentuados: 00 para MSX e 255 para ABNT. Depois disso, é só customizar os códigos de controle e o filtro está pronto. Mas, se a impressora a ser usada seguir a tabela ABICOMP ou qualquer outra que não tenha compatibilidade com o EXPERT, será o usuário quem definirá quais são os códigos dos caracteres acentuados (posi-

ção 128 em diante); como a parte de baixo da Tabela ASCII é igual às tabelas MSX e ABNT, a menção do filtro pode ser qualquer uma das duas.

Depois que uma das tabelas é escolhida, é mandatório digitar todos os labels listados no Manual, até o #14:. Se, como deve ser comum nestes casos, o usuário deseja definir quais as fontes de caracteres atribuídas às opções A, B, C e D, então os labels #15: a #22: devem ser preenchidos.

Vamos por etapas: os labels #01: a #04: são importantes terem seus valores o mais correto possível, em função dos tipos de fonte escolhidas, para permitir ao Editor um correto posicionamento do material impresso. Este labels contém o exato número de caracteres por linha que a impressora é capaz de imprimir de uma determinada fonte. Uma consulta ao manual da impressora geralmente revela estes valores, mas, na ausência dos mesmos, basta mandar a impressora imprimir uma linha inteira da fonte desejada e contá-los manualmente. De acordo com os valores destes labels, toda a vez que for escolhida uma dada fonte, o Editor alocará o espaço necessário na tela, de modo que o texto, quando impresso, fique corretamente justificado à direita ou centrado. Este recurso do ASTEX, diga-se de passagem, está presente em poucos editores de texto, mesmo dentre aqueles do PC. É importante que se faça um teste de validade destes valores. Na Figura 2, onde aparece um filtro que construí para a impressora Mônica 6011, o leitor poderá notar que a fonte C, de impressão condensada, teve seu valor redefinido de 132 para 101, enquanto que na fonte D, de impressão expandida, foi conservado o valor de 40 colunas, que é o verdadeiro para este tipo de impressão. Eu não sei explicar o porquê deste erro, mas, depois de umas contas mirabolantes, cheguei à conclusão de que deveria redefinir a impressão condensada para 101 caracteres (ou colunas), quando então a justificação pretendida foi alcançada.

Os valores dos labels acima mencionados afetam a impressão por redefinirem o número de colunas do Editor. Por isto, se um filtro for carregado e compilado com valores de coluna diferentes dos previstos para uma data fonte, o texto será automaticamente reformatado. Desta forma, é importante acertar um filtro de impressão e carregá-lo no Editor antes de digitar um texto, se houver intenção de usar fontes de caracteres com "pitch" diferentes (condensado, expandido e suas variações).

Na parte do filtro que define o início e fim de cada fonte, é preciso prestar uma certa atenção nas manhas da sua impressora. A Mônica, por exemplo, só imprime expandido se for desativada a impressão em qualidade de carta. Assim, no label que define esta fonte (veja a Figura 2), de #18:, foi prevista esta desati-

Figura 2

```
#00:255
#01:80
#02:80
#03:101
#04:40
#10:"MONICA 6011 - ABICOMP"
#11:27,"@"
#12:13
#13:10
#14:08
#15:27,"H"
#19:27,"G"
#16:27,"G"
#20:27,"H"
#17:15
#21:18
#18:27,"H",14
#22:27,"G",20

#26:27,"4"
#51:27,"5"
#31:27,"G",27,"7"
#56:27,"6",27,"H"
#36:27,"-",1
#61:27,"-",0
#41:27,"S",0
#66:27,"T"
#46:27,"S",1
#71:27,"T"

#27:27,"4"

#52:27,"5"
#32:27,"7"
#57:27,"6"
#37:27,"-",1
#62:27,"-",0
#42:27,"S",0
#67:27,"T"
#47:27,"S",1
#72:27,"T"
#28:27,"4"
#53:27,"5"
#33:27,"G",27,"7"
#58:27,"6",27,"H"
#38:27,"-",1
#63:27,"-",0
#43:27,"S",0
#68:27,"T"
#48:27,"S",1
#73:27,"T"

#29:20,27,"4"
#54:27,"5",14
#34:20,27,"G",27,"7"
#59:27,"6",27,"H",14
#39:20,27,"-",1
#64:27,"-",0,14
#44:20,27,"S",0
#69:27,"T",14
#49:20,27,"S",1
#74:27,"T",14

ç:198
Ç:166
É:168
Ê:169
À:161
Á:162
Â:163
Ã:164
Í:172
Ó:177
Ô:178
Õ:179
Ú:183
ó:209
ô:210
õ:211
à:193
á:194
â:195
ã:196
é:200
ê:201
í:204
ú:215
ª:220
º:221
°:191
§:190
~:126
#99:0
```

Listagem do filtro de impressão construído para uma impressora MONICA EI6011, eprom Versão D. As explicações sobre este filtro estão no corpo do texto. Fontes escolhidas: A – qualidade de dados; B – qualidade de carta; C – impressão condensada e D – impressão expandida.

vação, antes de enviar os códigos para impressão expandida. O ASTEX permite a inserção de quantos códigos forem necessários, bastando que os mesmos sejam separados entre virgulas.

O usuário do ASTEX poderá usar vários conjuntos de habilitadores e desabilitadores de fontes, sendo que um deles, definido na página 69 do Manual, serve para todas as fontes. Eu preferi e recomendo, definir esta seqüência de códigos para cada uma das fontes, embora com um pouco mais de trabalho de digitação. Fazendo isto, foi possivel otimizar a seqüência de códigos ideal para cada tipo de fonte, de modo a obter corretamente índice, expoente, sublinhado, etc. Como a Mônica não possui itálico, no seu lugar eu coloquei os códigos necessários para imprimir a fonte de caracteres alternativa, armazenada na RAM da impressora num dado momento.

A segunda parte do filtro, para customização dos caracleres acentuados, mostrada na Figura 2, refere-se à tabela ABICOMP. Note-se que só é preciso digitar o caracter em questão, sem necessidade de digitar a cerquilha ("#") que anteriormente identificava todos os labels Qualquer seqüência de Bytes poderá ser implementada para alcançar este objetivo. Uma dica que me foi dada pelo Julio Velloso é atribuir a caracteres da tabela MSX que não aparecem normalmente na impressora os códigos necessários para que o caracter possa ser impresso **graficamente**. Isto implica em digitar longas strings ativando os recursos gráficos, normalmente, no caso das impressoras nacionais, baseados no padrão Epson.

COMENTÁRIOS FINAIS:

A melhor maneira de aprender a trabalhar com um Processador de Textos é usando todos os seus recursos. Peço desculpas aos leitores da CPU por ainda não conseguir fazer uma avaliação mais profunda deste Editor. Pessoalmente, achei o ASTEX uma grande promessa como sendo o melhor Processador de Textos para o MSX. Só não sei se terá inicialmente um grande público, pois o programa ficará restrito aos que adquirirem a placa CT-80E. Sou frontalmente contra a inclusão de um aplicativo importante dentro de um cartucho. Nem o medo pela famigerada pirataria que flagela o padrão MSX no Brasil justifica tal atitude, como quizeram me fazer crer pessoas ligadas á Gradiente. No meu humilde entender, as pessoas da Gradiente deviam se preocupar com o controle de qualidade e evolução de seus produtos e portanto com o consumidor, que no caso, é usuário de um computador MSX defasado, mal montado e sem recursos. Estes novos modelos, por exemplo, não rodam a grande maioria dos programas (principalmente jogos) em poder atualmente dos usuários. Isto, por si só, desestimula vender o nosso 1.1 para atualizar o equipamento, já que nem isto é possível, pois se trata do mesmo MSX, com uma cosmética diferente. Igualmente inútil será convencer os usuários a não comprar programas adaptados pelas solthouses, pois, se por um lado a comercialização destes softwares é ilegal, por outro lado, foram eles próprios os responsáveis pela enorme venda dos MSX no Brasil.

Assim, creio que o ASTEX, a despeito dos excelentes recursos de edição, fica prejudicado por ser disponível apenas embutido na placa de 80 colunas. Usuários do MSX 2.0, que poderiam usufruir deste programa, ficarão, portanto, a ver a banda passar, pelo menos, por enquanto, enquanto o "seu" lobo não vem.

CASTLE II
Parte II

Neste artigo vamos dar a seqüência e as dicas para conseguirmos a primeira CHAVE laranja.

CARLOS DOS SANTOS

2ª SEQÜÊNCIA PARA CONSEGUIR A 1ª CHAVE LARANJA

E9 – F9 – G9 – F9 – G9 – H9 – H8 – I8 – J8 – J7 – I7 – H7
G7 – F7 – G7 – H7 – G7 – H7 – I7 – J7 – J8 – I8 – H8 – H9
H8 – G8 – F8 – G8 – H8 – H9 – H8 – I8 – J8 – J9 – I9 – I8

OBS.: A cada vez que você tiver que sair de uma sala, verifique no mapa "AUTÊNTICO", qual a melhor saída. Na falta do mapa, sugerimos que antes de sair da sala, grave na fita a situação atual. É muito fácil você se enganar de salda e não poder mais voltar ou não ter mais chaves para as outras portas ou, simplesmente, se perder.

DICAS ESPECIAIS

Sala F8 – Muita velocidade, muita atenção e obsevem bem a ordem dos movimentos para cada SACO (veja figura da sala resolvida).

Sala F9 – Muita habilidade será necessária. Suba só até o

SALA F8 – RESOLVIDA.

quarto TRANSPORTE FLUTUANTE (de baixo para cima), pegue uma carona com o terceiro, pegue as CHAVES e suba até a PORTA.

Sala G7 – A cortina do ARCO-ÍRIS é fundamental para, com o auxílio dos TIJOLINHOS, destruir todos os REIZINHOS e fugir do FOGUINHO na sala ao lado.

Sala G8 – A primeira coisa a fazer é colocar metade do TIJOLINHO na caçamba do GUINDASTE e depois resolva o quebra-cabeça (veja figura da sala resolvida).

Sala G9 – Para pegar todas as CHAVES com segurança, destrua os inimigos com o TIJOLINHO e depois recoloque o TIJOLINHO no seu lugar de origem (tal como explicamos no primeiro artigo). Tenha muita paciência. Esta sala é demorada mesmo mas é tranqüila, sem nada para importuná-lo (veja figura da sala resolvida).

Sala H7 – Tudo é feito com a proteção da cortina do ARCO-ÍRIS.

Sala H9 – Trate de destruir os dois SOLDADOS dos níveis superiores, pois você vai precisar destas passagens mais tarde.

SALA G8 – RESOLVIDA.

SALA G9 – RESOLVIDA.

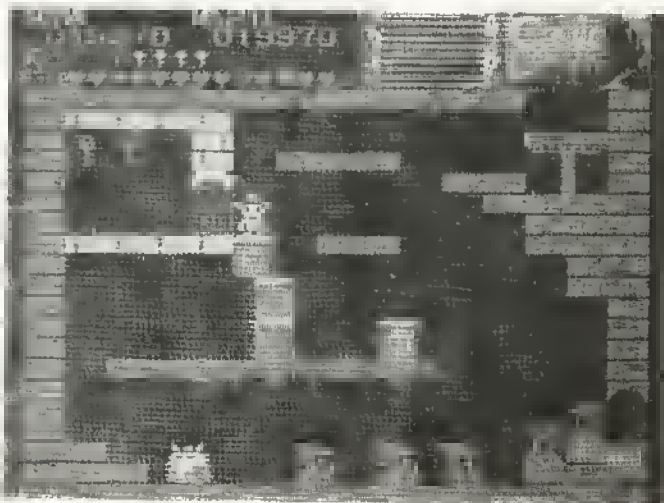
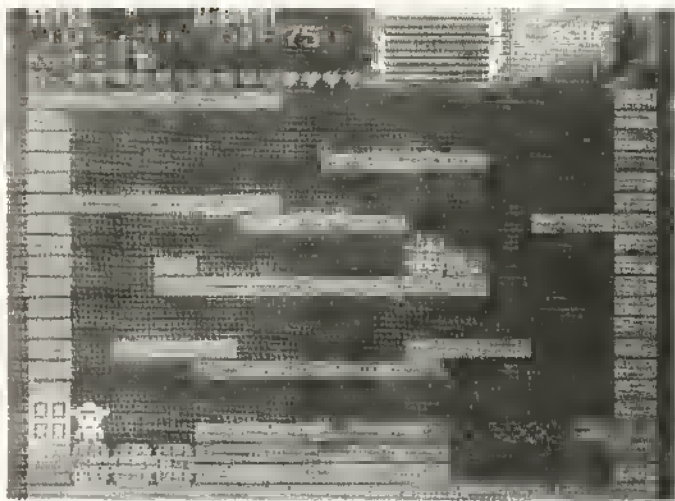
Sala I7 – Cuidado, às vezes, conseguir uma vida extra, sai caro.

Sala I8 – Fuja dos REIZINHOS. Derrubando o TIJOLINHO, o TRANSPORTE FLUTUANTE vai até a PORTA.

Sala I9 – Passe correndo que as BRUXINHAS dão tempo. Destrua um REIZINHO. O REIZINHO só desce um degrau, se você estiver num piso mais abaixo dele (use o salto demorado). O posicionamento de cada TIJOLINHO é fundamental para você poder empurrar um TIJOLINHO sobre o outro. Entre os dois TIJOLINHOS que ficam no piso, tem que haver sempre um espaço para o PRÍNCIPE entrar (veja figura da sala resolvida).

Sala J7 – Muita paciência e precisão nos movimentos. O momento certo de um salto tem que estar combinado com o momento em que a "FLORZINHA" está dormindo.

Sala J8 – Assim como você pega carona quando está por cima, pegue por baixo também.

Sala J9 – A primeira coisa a fazer é destruir o REIZINHO na ONDINHA mas, para isto, você só pode pegar uma BARRA DE OURO. Tenha paciência porque o resto

SALA I9 – RESOLVIDA.

SALA J9 – RESOLVIDA

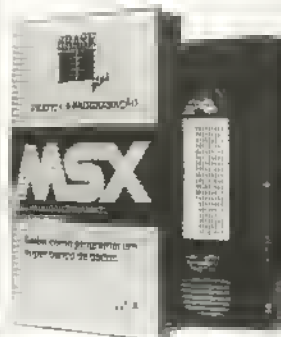
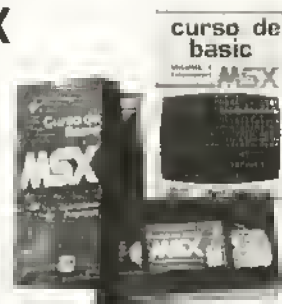
é demorado e mais uma coisa: o segundo BARRIL a ser levado para perto da PORTA, só pode cair naquele piso quando o terceiro BARRIL estiver no meio da ONDINHA (veja figura da sala resolvida).

Após a última parte do artigo, quando daremos as dicas para você conseguir libertar a PRINCESINHA, se você nos descrever o que acontece quando a PRINCESINHA é libertada (como é a tela que aparecerá), junto com as "senhas" de cada artigo e uma palavra-chave escondida nas revistas dos artigos, você receberá com exclusividade num diskete (não serão impressos na revista), os programas que foram usados para se extrair cada um destes desenhos que são operados pelo jogo como SCREEN1 e foram transformados em SCREEN2 para poderem ser impressos. A senha deste artigo, é a palavra paciência. Colecione as revistas pois as palavras-chaves necessárias, só serão reveladas na edição da última parte do artigo.

Boa sorte e até o mês que vem.

SEQÜÊNCIAS SOLICITADAS

(Diagrama: SALA E8, com os pontos a, b, c)

1 – Caminhe sem parar para a direita até ficar exatamente do lado direito de onde vai cair o primeiro TIJOLINHO.
2 – Assim que o TIJOLINHO terminar de cair, empurre-o para a esquerda apenas um passo e,
3 – Imediatamente, salte-o, posicionando-se exatamente à esquerda de onde vai cair o segundo TIJOLINHO.
4 – Assim que o segundo TIJOLINHO terminar de cair, empurre-o para a direita apenas um passo e pronto.
5 – Espere o terceiro TIJOLINHO cair e está feita e escadinha para você alcançar o piso das ESTEIRAS E OUTRAS SALAS...

WONDER BOY

André Luis Anciães dos Santos

Neste jogo, você é um corajoso índio, que tem como objetivo salvar a sua amada do malvado mago que a raptou. Para isso, você terá que passar pelos perigos da selva, como aranhas venenosas, sapos e borboletas assassinas e outros seres. A selva está dividida em quatro partes (fases), cujos mapas estão publicados aqui. Abaixo, vão algumas dicas sobre cada fase, para facilitar a missão do seu índio.

FASE 1

Esta fase é a mais fácil de todas. Não há nenhuma dica nela.

FASE 2

Esta fase é marcada pela necessidade de pular bastante.

Sempre que você estiver passando por um local onde existem polvos, procure correr sempre, sem parar, pois assim eles não o atingirão.

No local onde há um lobo desenhado, faça o seguinte: assim que cair no chão, use o pulo maior (veja instruções de jogo), e, ao cair novamente, atire. Isso evitará que o lobo o pegue.

FASE 3

Não há maiores dicas nesta fase. Apenas, quando for matar os morcegos, ande aos poucos, para que venha um de cada vez.

Logo após os morcegos, para matar os sapos, faça o seguinte: primeiro, antes de pular para subir no lugar onde estão os sapos, mate o da frente, pulando parado e atirando. Então, suba e mate o outro.

FASE 4

Logo no início, onde existem as aranhas, faça o seguinte:

– Mate todas as aranhas;
– Vá andando devagar até aparecer a primeira borboleta. Mate-a;
– Avance até a fogueira. Preste atenção, pois o lobo irá aparecer quando você se aproximar da fogueira. Assim que ele aparecer, salte;

Ao final, quando for lutar com o monstro, preocupe-se apenas com os tiros dele. Atire sem parar.

INSTRUÇÕES DE JOGO

SETA P/DIREITA – MOVE PARA A DIREITA

SETA P/ESQUERDA – MOVE P/ESQUERDA

Z – PULA

X – ATIRA

SETA P/O LADO + SETA P/CIMA + Z – ATIRA E PULA MAIS ALTO

SÍMBOLOS

1º grupo: BÔNUS

OVO – Pode fornecer, se destruído, quatro diferentes bônus.

 – Fornece imortalidade temporária.

 – SKATE – Permite a você esbarrar em um (e apenas um) inimigo.

 – PEIXE – Ao final da fase, lhe dará pontos extras.

 – LEITE – Faz sua energia voltar ao máximo.

2º grupo: ENERGIA

 – BANANA

 – MAÇÃ

3º grupo: PERDA DE ENERGIA

 – PEDRA

4º grupo: MORTE

 – BORBOLETA

 – ARANHA

 – COBRA

 – SAPO

 – POLVO

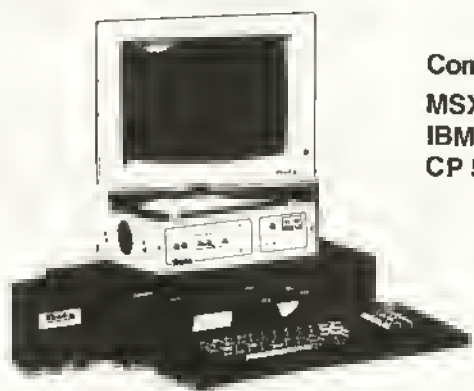
 – CARACOL

 – LOBO

 – MORCEGO

 – FOGUEIRA

5º grupo: MOVIMENTAÇÃO

 – MOLA – Joga você para o alto.

COBRA'S ARK

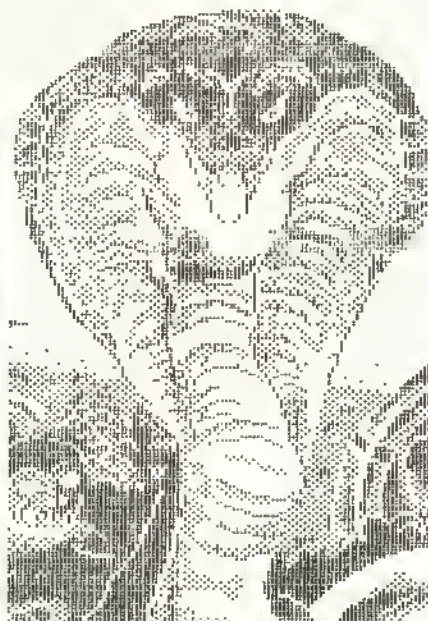
Em mais este jogo da Dinamic, você será um cavaleiro medieval e seu objetivo será conseguir a arca do Príncipe Cobra, que guarda valiosos tesouros. Para isso, você contará com a ajuda de uma feiticeira e anões, lutando com as figuras mais estranhas.

André Luis Anciães dos Santos

O MAPA

Há um mapa do jogo publicado em outra página. No mapa, os símbolos são os seguintes:

PERSONAGENS:
A – SIVILIX (ANÃO PÚRPURA)
V – SERVAX (ANÃO VERDE)
G – GRAUDIER (FILÓSOFO)
D – DRAGÕES
B – GURKA (BÁRBARO)
F – DRAGORA (FEITICEIRA)
P – PRÍNCIPE COBRA
X – FEITICEIRO DO TEMPLO

OBJETO:
S – SILEX
C – COPA DE OURO
J – BALSA
E – ESPADA
R – RELÓGIO DE AREIA

Alguns símbolos não estão no mapa. Isso se deve ao fato destes símbolos representarem objetos ou personagens que não tem localização exata. Dos símbolos que estão no mapa, apenas os seguintes têm localização exata: G (Graudier), F (Dragora), D (Dragões), P (Príncipe Cobra), X (Feiticeiro do Templo), J (Balsa). Os outros objetos desenhados no mapa estão localizados na posição em que mais aparecem, porém podem estar em qualquer outro lugar.

As letras I (quadrados A8, D8 e G8) marcam os três locais em que o jogo pode iniciar, sendo que isso é decidido no início, por sorteio.

Em cada quadrado, há um retângulo com uma palavra que descreve o lugar, para uma melhor localização no mapa. As setas, entre os quadrados, indicam as direções em que você pode se locomover. As portas indicam Entrada/Saída, sendo que a porta 1 se comunica com a outra porta 1, e a porta 2 com a outra 2.

As duas partes separadas do mapa representam o interior do Castelo de

Glouster e do Templo Mágico. O acesso a elas é feito através das portas 1 e 2, localizadas nos quadrados G3 e D3 respectivamente. Estando nesses quadrados, mande-os **entrar** (veja parte de **comandos**). Para sair do castelo ou do templo, é só ir até á primeira sala dos mesmos e dar o comando **sair**.

O JOGO

O mapa é dividido ao meio pelo mar (linha 5). Observe que após passar para a parte de cima do mapa, você não poderá mais voltar para a parte de baixo. Portanto, antes de passar para a parte de cima, você deve pegar todos os objetos existentes na parte de baixo. Na parte de baixo, existem 2 objetos: um deles é a balsa (D6), e o outro é aleatório, assim como a localização.

Após pegar a balsa e o outro objeto, passe para a parte de cima, passando pelo quadrado A5. Abaixo estão as funções dos objetos:

BALSA – Permite que você ande pelo mar. Sem ela, você morre ao entrar nele.
COPA DE OURO – Usada para matar o Príncipe Cobra.
ESPADA – Mata os dragões e os personagens em geral.
SILEX – Mata o Feiticeiro.
RELÓGIO DE AREIA – Deve ser entregue à feiticeira, para que ela lhe dê a chave do templo cobra.

Abaixo, está um roteiro para terminar o jogo:

– Vá até o quadrado D6 e pegue a balsa.
– Procure o outro objeto que fica na parte de baixo do mapa.
– Passe para a parte de cima.
– Procure os outros objetos (inclusive no Templo Mágico e no Castelo Glouster), evitando entrar nos quadrados G1 e G4 e nos quadrados onde estão os dragões.
– Vá até a feiticeira (E4). Ela lhe dará a chave do Templo Cobra.
– Vá até o Príncipe Cobra (G1). Ele lhe tomará a Copa de Ouro, beberá o conteúdo e morrerá.
– Vá até os dragões e mate-os, com a espada.
– Vá até o Templo Cobra. Use a chave para abrir a porta e entre. O jogo terminará.

DICAS

– No caminho, mate, se encontrar, Graudier e Gurka.
– Só vá até o Príncipe Cobra, o Feiticeiro e os Dragões depois que tiver pego todos os objetos.
– Após conseguir a chave, evite encontrar Gurka, ou ele a tirará de você. Para recuperá-la, você terá de encontrá-lo e matá-lo.
– Os quadrados marcados com cruzes indicam morte.
– Não dê ouvidos aos conselhos de Graudier.
– Não mate os anões nem a feiticeira.

CONTROLES

Os movimentos, neste jogo, são dados através de um cursor, que passa sobre os comandos. Os comandos são representados por desenhos, que estão desenhados em outra página. Abaixo, o significado de cada um, pelo seu número:

1 – OESTE – Move você para Oeste.
2 – NORTE – Move você para Norte.
3 – SUL –Move você para Sul.
4 – LESTE – Move você para Leste.
5 – LUPA – Serve para verificar qual o objeto que está no local, se existir algum.
6 – FITA – Salva o jogo para posterior continuação ou carrega um jogo que já esteja gravado. Existem opções para Disco e Fita.
7 – STOP – Pára o jogo e pergunta se você deseja continuar ou abortar.
8 – COGER – Pega um objeto que esteja no local, se houver.
9 – PODERES MÁGICOS –Dá dicas sobre o que fazer.
10 – USAR – Serve para usar o objeto. Por exemplo, para matar o Feiticeiro, vá até ele, coloque em usar e aperte o botão. Aparecerá um outro cursor sobre os objetos. Coloque o cursor sôbre o Silex e aperte o botão. O Feiticeiro morrerá. Proceda de maneira análoga para matar os outros personagens (usando a espada) e para abrir a porta do Templo Cobra (usando a chave).
11 – BÚSSOLA – Dá a direção em que se encontra o Templo Cobra.
12 – ENTRAR – Serve para entrar no Templo Cobra, no Templo Mágico e no Castelo Glouster.
13 – SAIR – Serve para sair dos locais acima, desde que você esteja na sala inicial.
14 – VOZ – Liga/Desliga sintetização de voz.

COBRA'S ARK
A
B
C
D
E
F
G
1
2
3
4
5
6
7
8
CORREDOR
CORREDOR
CORREDOR
TEMPLO COBRA
CORREDOR
CAVERNA P
CORREDOR V
CORREDOR G
TEMPLO D
CORREDOR
CAVERNA
CAVERNA
CORREDOR
CORREDOR B
GLOUSTER 2
CAVERNA
CORREDOR
TEMPLO MÁGICO 1
CORREDOR
CORREDOR
CORREDOR
CORREDOR
CORREDOR F
CORREDOR
CAVERNA X
MAR
MAR
MAR
MAR
MAR
MAR
CORDILHEIRA
CORDILHEIRA
CORDILHEIRA
ORILLA J
CORDILHEIRA
CORDILHEIRA
DESERTO
CORDILHEIRA
MAR
CORDILHEIRA
CORDILHEIRA
DESERTO
MAR
DESERTO
HIMALAIA
MAR
CIDADE MURADA
DESERTO
MAR
DESERTO
ENTRADA 1
CORREDOR
CORREDOR
CORREDOR
CORREDOR
CAVERNA
TEMPLO MÁGICO
CORREDOR 2
CORREDOR
CORREDOR
CORREDOR
CORREDOR D
CASTELO GLOUSTER
N
S
O
L

GONZZALEZZ II

Leonardo de Lima

INTRODUÇÃO

Este e mais um magnífico jogo da ÓPERA SOFT.

O jogo possui bom enredo e belas cenas típicas do México.

O JOGO

O jogo pode ser dividido em 9 (nove) fases diferentes. São elas:

- FASE DOS CACTOS
- FASE DAS TENDAS
- FASE DOS CATOS 2
- FASE DA FAZENDA
- FASE DAS PEDRAS
- FASE DAS TENDAS 2
- FASE DAS PEDRAS 2
- FASE DA FAZENDA 2
- FASE DOS CACTOS 3

Obs.: As fases foram representadas por nomes que eu mesmo dei, para facilitar a sua localização.

O jogo possui também 8 (oito) objetos diferentes. São eles:

REVOLVER

Serve para matar os inimigos. A arma deve ser carregada para poder ser utilizada.

FACA

Idem revolver, porém tem alcance menor e nunca acaba.

CACHIMBO

Servirá para consolidar a paz entre você e os índios.

BARRA DE ENERGIA

Irá repor a sua energia ao ser solicitada.

BANCO DE DINHEIRO

Servirá para pagar os músicos.

MUNIÇÃO

Irá recarregar o revólver.

Obs.: Não encontrei função para o extintor e a garrafa.

AS FASES

1ª – FASE DOS CACTOS, O INÍCIO

Nela você terá que pegar o revólver, pular a ponte quebrada e pegar a faca.

2ª – FASE DAS TENDAS

Você terá que travar batalhas com índios e escorpiões nesta fase. Não há objetos.

3ª – FASE DOS CACTOS 2

Este "2" significa que esta é a 2ª fase dos cactos que você encontrará. Isto vale para as outras fases.

Você terá que matar os bichinhos que aparecerão nesta facílima fase.

4ª – FASE DAS FAZENDAS

Nesta fase você encontrará a garrafa, o extintor e o cachimbo. Somente pegue o cachimbo para posterior utilização.

Você poderá entrar em uma casa que está com a porta aberta. É nela que você encontrará o extintor e cachimbo escondidos em dois cestos. Cuidado com os inimigos!!!

5ª – FASE DAS PEDRAS

Esta é uma das fases mais simples, porém chata. Não há objetos.

6ª – FASE DAS TENDAS 2

Você poderá entrar na última tenda desta fase. Nela você encontrará uma barra que lhe dará energia, e um saco de dinheiro para posterior utilização. Você poderá também fumar o cachimbo (da paz) com o índio presente, pressionando a barra de espaço.

7ª – FASE DAS PEDRAS 2

Nesta fase você encontrará uma nova ponte que deverá ser atravessada. Bem no final desta fase, começará a tocar uma música que tirará sua energia rapidamente. Quando você ouvir a musiquinha começar a tocar, vá rapidamente em direção de uma mulher que estará dançando, e dê o saco de dinheiro (apertando a barra de espaço) que tudo voltará à normalidade.

8ª – FASE DAS FAZENDAS 2

Para mim foi a mais difícil. Após a mulher parar de dançar, siga em frente tomando cuidado com as minhocas e os "homenzinhos" que lhe dão rasteira.

Você poderá entrar em outra casa (o bar), onde achará outro saco de dinheiro que será vital para a conclusão do jogo.

Após pegar o saco, saia do bar e siga em frente. Logo em seguida você encontrará outro "cantante" que deverá ser devidamente remunerado pelo seu espetáculo (remunere com o saco de dinheiro).

9ª – FASE DOS CACTOS 3

Esta não é bem uma fase. Esta é uma parte do jogo dedicada a seu final. Agora que você encontrou a sua rede predileta, deite-se e descanse, mas tome bastante cuidado com "LAS PESADILLAR!!!".

COMANDOS

Além dos quais o jogo apresenta, as setas cursoras também funcionam.

Para selecionar o objeto a ser usado, você deverá usar os nº de 1 a 4, respectivamente.

Para entrar nas habitações, basta pressionar a tecla cursora para baixo ou o equivalente.

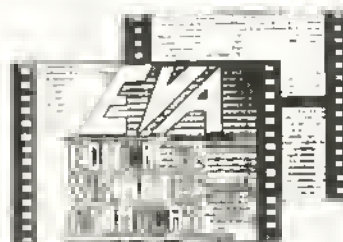
Caso você obtenha alguma dúvida neste jogo, escreva para:

Leonardo de L. B. Lima
Av. Ary Parreiras, 432
CEP: 24230
Vital Brazil – Niterói – R.J.

Tudo para o seu MSX

DESPACHAMOS PARA TODO O BRASIL

Conheça a sofisticação e magia do Sistema Gráfico Aquarela.
— Recursos completos para edição de telas gráficas.
— Figuras prontas para você usar e ilustrar suas telas.
— Caracteres em Out-Line, bold, sombra no tamanho 8x8 ou 16x16.

LADY 80 Temos Impressora Elgin Lady 80, A Pequena Notável.

E.V.A.
EDITOR DE VINHETAS ANIMADAS

PROMOÇÃO: KIT 2.0 DDX

Nemesis: Top-Cad - MSX Computer Aided Design, ou Desenho Auxiliado por Computador. Mais uma novidade em software profissional que a Nemesis desenvolveu especialmente para o seu MSX!

JOGOS
NOVIDADES: Temos os últimos lançamentos.

PACOTES: Na compra de 50, 100 e 200 jogos, uma super oferta. Consulte-nos ou peça catálogo grátis.

PERIFÉRICOS: Linha completa de periféricos para MSX e PC.

PROMOÇÃO: Na compra de um drive, você ganha um brinde especial. Preços abaixo do mercado.

SUPRIMENTOS: Capas - Porta Disquetes - Disquetes - Livros Específicos para MSX (Preços promocionais com 15% de desconto) - Fitas para Impressoras.

SOFTWARE: Educativos - Aplicativos - Utilitários - Desenvolvemos também sistemas específicos para empresas.

KIT PARA DRIVE DDX: Composto de gabinete, metálico com fonte de alimentação, interface DDX com cabo de ligação para dois drives, sistema operacional (DOS) e manual.

MEGARAM DISK 256: Placa de expansão de memória de 256 Kbytes, com o sistema operacional DDX DOS residente.

Game of Time
SOFTWARE E HARDWARE

Av. [illegible]
Paulo - SP - **Fone:** (011) 581-2739

CLUBE DO LEITOR
O CARTÃO DO MSX

MANÍACOS DO MSX
15% desconto na compra de jogos.
20% desconto na compra de jogos especiais.
10% desconto na compra de programas de autores nacionais.
15% desconto na compra de aplicativos.

CONECTOR IND. E COM. LTDA
5% desconto na compra de kit de drive para MSX.

CIBERTRON ELETRÔNICA LTDA
15% desconto na compra de software.

YOUNGSOFT
30% desconto nas compras de software.
10% desconto na inscrição no clube de usuários.

NEMESIS INFORMÁTICA LTDA
10% desconto em seus produtos.

RECURSOS DIGITAIS
5% desconto na compra de periféricos.
10% desconto na compra de softs de outras empresas.
30% desconto na compra de softs da Redi Universoft.

TACTO INFORMÁTICA COM. LTDA.
10% desconto na compra de qualquer produto ou curso.

PAULISOFT INFORMÁTICA LTDA
10% desconto na compra de software, exceto promoção.

DISCOVERY INFORMÁTICA
10% desconto em seus produtos.

EDITORA ALEPH
15% desconto em suas publicações.

REVOLUTION
20% desconto nas compras de software.

NEWSOFT
10% desconto na compra dos jogos comuns.
20% desconto nos jogos especiais.
25% desconto nos aplicativos.
30% desconto na compra de livros.
5% desconto na compra de periféricos e suprimentos.
* Os descontos acima não incidirão sobre produtos em promoção.

THUNDERSOFT
20% desconto em todos os seus produtos.

NEWDATA
5% desconto nos produtos de representação/revenda.
10% desconto nos seus produtos.

ESPACIAL ELETRÔNICA
20% desconto nos seus produtos.

INFORTELLES
5% desconto em geral.

GAME OF TIME
10% desconto em geral.

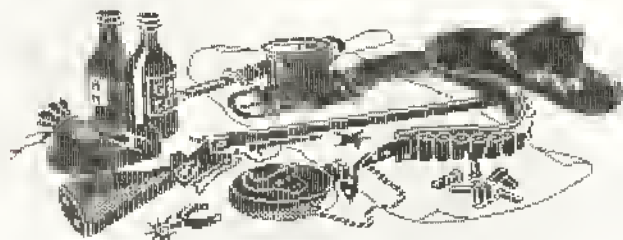
SOFTMARK
12% desconto nos seus produtos.

SOFT DESIGNS
15% desconto na compra de software e serviços.

MSX INFORMÁTICA
10% desconto em hardware.
20% desconto em software da MSX INFORMÁTICA e ou
10% desconto em software de outras EMPRESAS.
10% desconto em assistência técnica e suprimentos.

Se você ainda não tem um cartão,
faça logo a sua assinatura de CPU e receba o seu!

GUN FRIGHT

Como você se sentiria se fosse um xerife de uma cidade do velho oeste americano? Certamente, as respostas para essa pergunta são as mais variadas possíveis. Mas, seja qual for a sua, você gostará deste jogo.

André Luis Anciães dos Santos

O JOGO

Gun Fright consiste em uma caça aos bandidos mais temidos e procurados do velho oeste. Você deverá caçá-los um por vez, para receber as recompensas. Seu dinheiro serve para alugar cavalos, comprar balas para seu Colt e indenizar as possíveis vítimas inocentes de sua caçada.

No início do jogo, você deverá guiar uma mira pela tela para procurar acertar os sacos de dinheiro que caem. Quanto mais sacos você acertar, mais dinheiro você terá no início do jogo.

Quando acabarem os sacos, o jogo em si terá início. Você aparecerá em frente à delegacia, para onde você voltará sempre que perder uma de suas vidas.

No canto esquerdo superior aparecerá a fotografia do bandido que você deve perseguir, assim como seu nome e a recompensa oferecida. A partir daí, você deverá andar pela cidade à procura dele, se guiando pelos "dedo-duros" e evitando as senhoras, sempre que puder. Mas isso já é papo para outro item.

OS PERSONAGENS

Os personagens do jogo são os seguintes:

- VOCÊ (O xerife pomposo e forte);
- AS SENHORAS (As frágeis donzelas);
- OS "DEDO-DURO" (O baixinho saltitante);
- OS BANDIDOS (O feio e cruel vilão).

Obviamente, o xerife dispensa comentários. Afinal, é conhecido em todo o oeste.

As senhoras têm um objetivo bem definido no jogo: atrapalhar você. Elas andam pela cidade todo o tempo. Se você esbarrar em uma delas, perde uma vida. Você pode atirar nelas, porém perderá dinheiro (O valor varia, e fica escrito na parte inferior direita da tela, ao lado da palavra FINE).

Os "DEDO-DURO" estão espalhados pela cidade, e não se movimentam. Eles servem para ajudá-lo a encontrar o bandido, apontando para a direção em que ele se encontra.

OBS.: O "DEDO-DURO" não aponta exatamente para a direção onde o bandido se encontra, e sim na direção que apresenta a menor distância do ponto onde ele se encontra.

Os BANDIDOS são o que você procura. Eles podem estar em diversos lugares, dentro de casas, galpões, e se movimentam todo o tempo.

COMO JOGAR

Você deve sempre procurar os DEDO-DURO, e seguir na direção por ele indicada, mas usando também o bom senso, ou seja, se ele indicar uma direção e você andar nessa direção por algum tempo e não encontrar o bandido, isso quer dizer que ele deve estar pelo caminho, um pouco fora da direção que foi indicada.

Os cactos matam. Afaste-se deles.

Se você é daqueles que gosta de fazer muitos pontos, não deve usar o cavalo. Porém, se você quer conseguir matar os bandidos, não deixe de pegar os cavalos.

Quando você encontrar o bandido, atire nele. Quando ele receber seu tiro, tocará uma música e aparecerá uma tela com uma mira e um homem de frente para você. Seja rápido, pois o bandido é um ótimo pistoleiro. Coloque a mira em qualquer parte do corpo do bandido e atire. Se você conseguir matá-lo, aparecerá o próximo bandido para você procurar. Caso contrário, você perderá uma vida.

TECLAS

SHIFT – PAUSA E REINÍCIO
RETURN – MUDA O ÂNGULO DE VISÃO (EXPERIMENTE)
SETAS OU JOYSTICK – MOVEM
ESPAÇO OU BOTÃO – ATIRA

DICAS

– Procure não matar as senhoras, mas se for necessário não hesite em fazê-lo.

– Sempre que for acertar o bandido para ir para o duelo, certifique-se de ter pelo menos três balas. Se tiver menos, gaste-as, para evitar perda de tempo para recarregar durante o duelo.

– Se você estiver a cavalo, matará as senhoras e os dedo-duro ao esbarrar neles.

– Os dedo-duros também matam se você encostar neles, e morrem se atingidos por um tiro.

– Alguns bandidos vêm a cavalo, portanto cuidado.

– Não tenha pressa. A pressa pode ser fatal.

GUN FRIGHT

O MUNDO É DOS EXPERTS.

Quer dizer que você quer ser o melhor? Quer entrar no futuro pela porta da frente? Chegaram os novos Experts da Gradiente, os primeiros micros para quem entende e para quem não entende de computador. Para quem entende, o Expert DD Plus é o único no Brasil com disk-drive de 3,5 polegadas embutido. Para quem não entende, o Expert Plus é muito fácil de operar e até vem com um programa na memória que mostra um pouco de tudo o que você pode fazer com ele. Isso significa que é só comprar e começar a usar. Os Experts têm disponíveis todos os periféricos necessários para o seu sistema não parar de crescer, como o Multi-Modem, os Cartões 80 Colunas e muitos outros. Sem falar nos 1500 programas para você fazer desde gráficos até música, do controle financeiro de uma pequena empresa até jogos da loto, aprender francês, controlar o saldo do banco ou escrever um livro. Com um Expert você tem acesso ao Videotexto, a outros Experts via linha telefônica e a um processador de textos que vai deixar você sem palavras. Ele é agenda, arquivo, central de informações, video game e até terminal de PC. Para conhecer melhor os novos micros da Gradiente, peça uma demonstração. Afinal, hoje você é expert ou não é nada.

Produzido na Zona Franca de Manaus